User Guide for Outposts servers

AWS Outposts

AWS Outposts: User Guide for Outposts servers

Table of Contents

What is AWS Outposts?

AWS Outposts is a fully managed service that extends AWS infrastructure, services, APIs, and tools to customer premises. By providing local access to AWS managed infrastructure, AWS Outposts enables customers to build and run applications on premises using the same programming interfaces as in AWS Regions, while using local compute and storage resources for lower latency and local data processing needs.

An Outpost is a pool of AWS compute and storage capacity deployed at a customer site. AWS operates, monitors, and manages this capacity as part of an AWS Region. You can create subnets on your Outpost and specify them when you create AWS resources such as EC2 instances and subnets. Instances in Outpost subnets communicate with other instances in the AWS Region using private IP addresses, all within the same VPC.

> **Note**
>
> You can't connect an Outpost to another Outpost or Local Zone that is within the same VPC.

For more information, see the AWS Outposts product page.

Key concepts

These are the key concepts for AWS Outposts.

- **Outpost site** – The customer-managed physical buildings where AWS will install your Outpost. A site must meet the facility, networking, and power requirements for your Outpost.
- **Outpost capacity** – Compute and storage resources available on the Outpost. You can view and manage the capacity for your Outpost from the AWS Outposts console. AWS Outposts supports self-service capacity management that you can define at the Outposts level to reconfigure all of the assets in an Outposts or specifically for each individual asset. An Outpost asset can be a single server within an Outposts rack or an Outposts server.
- **Outpost equipment** – Physical hardware that provides access to the AWS Outposts service. The hardware includes racks, servers, switches, and cabling owned and managed by AWS.

- **Outposts racks** – An Outpost form factor that is an industry-standard 42U rack. Outposts racks include rack-mountable servers, switches, a network patch panel, a power shelf and blank panels.
- **Outposts servers** – An Outpost form factor that is an industry-standard 1U or 2U server, which can be installed in a standard EIA-310D 19 compliant 4 post rack. Outposts servers provide local compute and networking services to sites that have limited space or smaller capacity requirements.
- **Outpost owner** – The account owner for the account that places the AWS Outposts order. After AWS engages with the customer, the owner may include additional points of contact. AWS will communicate with the contacts to clarify orders, installation appointments, and hardware maintenance and replacement. Contact AWS Support Center if the contact information changes.
- **Service link** – Network route that enables communication between your Outpost and its associated AWS Region. Each Outpost is an extension of an Availability Zone and its associated Region.
- **Local gateway (LGW)** – A logical interconnect virtual router that enables communication between an Outposts rack and your on-premises network.
- **Local network interface** – A network interface that enables communication from an Outposts server and your on-premises network.

AWS resources on Outposts

You can create the following resources on your Outpost to support low-latency workloads that must run in close proximity to on-premises data and applications:

Compute

Resource type	Racks	Servers
Amazon EC2 instances	 Y	 Yes
Amazon ECS clusters	 Y	 Yes

Resource type	Racks	Servers
Amazon EKS nodes	 Y	 No

Database and analytics

Resource type	Racks	Servers
Amazon ElastiCache nodes (Redis cluster, Memcached cluster)	 Y	 No
Amazon EMR clusters	 Y	 No
Amazon RDS DB instances	 Y	 No

Networking

Resource type	Racks	Servers
App Mesh Envoy proxy	 Y	 Yes
Application Load Balancers	 Y	 No

Resource type	Racks	Servers
Amazon VPC subnets	Y	Yes
Amazon Route 53	Y	No

Storage

Resource type	Racks	Servers
Amazon EBS volumes	Y	No
Amazon S3 buckets	Y	No

Other AWS services

Service	Racks	Servers
AWS IoT Greengrass	Y	Yes

Pricing

Pricing is based on your order details. When you place an order, you can choose from a variety of Outpost configurations, each providing a combination of Amazon EC2 instance types and storage options. You also choose a contract term and a payment option. Pricing includes the following:

- **Outposts racks** - Delivery, installation, infrastructure service maintenance, software patches and upgrades, and rack removal.
- **Outposts servers** - Delivery, infrastructure service maintenance, and software patches and upgrades. You are responsible for the installation and packing the server for return.

You are billed for shared resources and any data transfer from the AWS Region to the Outpost. You are also billed for data transfers that AWS performs to maintain availability and security.

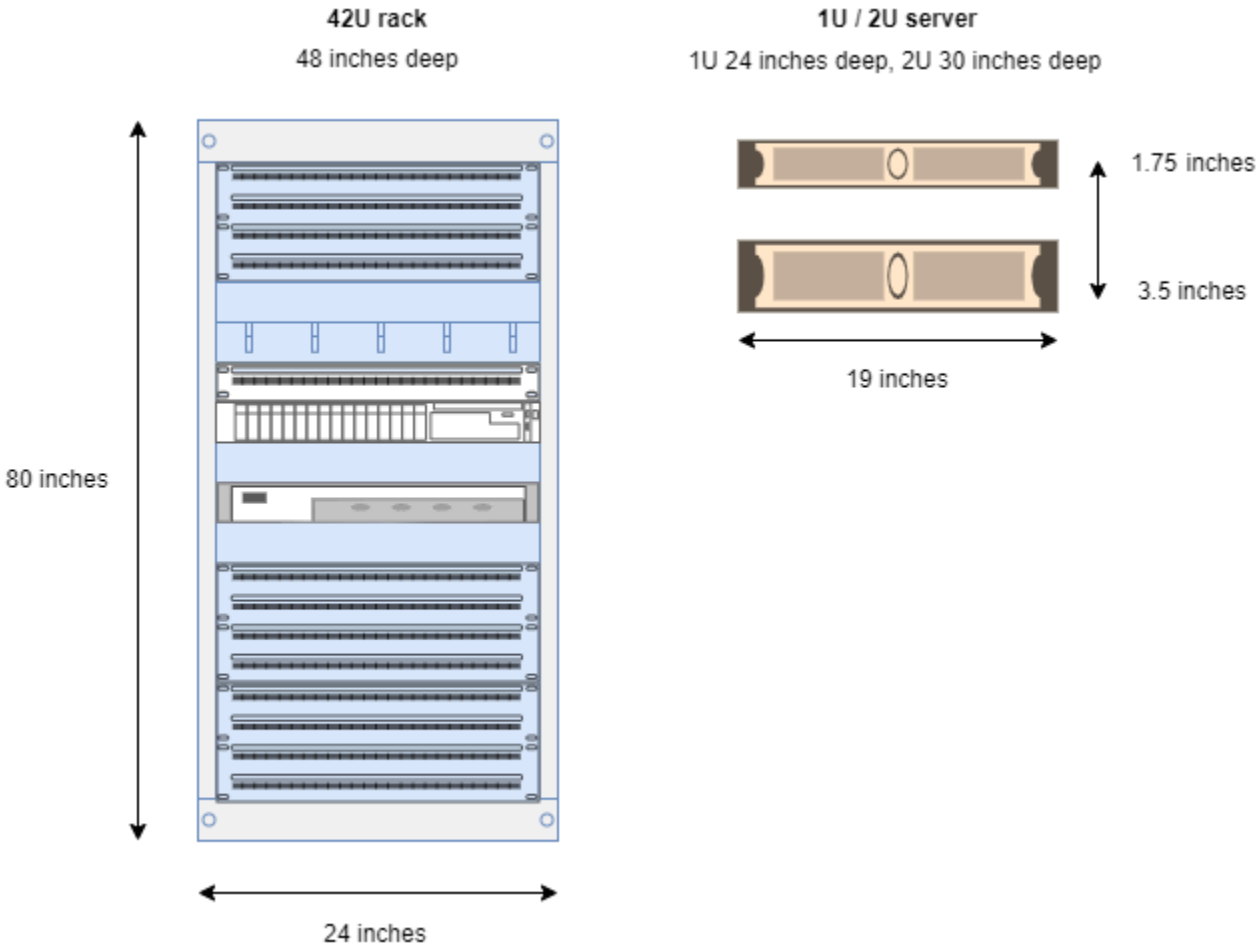
For pricing based on location, configuration, and payment option, see:

- Outposts racks pricing
- Outposts servers pricing

How AWS Outposts works

AWS Outposts is designed to operate with a constant and consistent connection between your Outpost and an AWS Region. To achieve this connection to the Region, and to the local workloads in your on-premises environment, you must connect your Outpost to your on-premises network. Your on-premises network must provide wide area network (WAN) access back to the Region. It must also provide LAN or WAN access to the local network where your on-premises workloads or applications reside.

The following diagram illustrates both Outpost form factors.



Contents

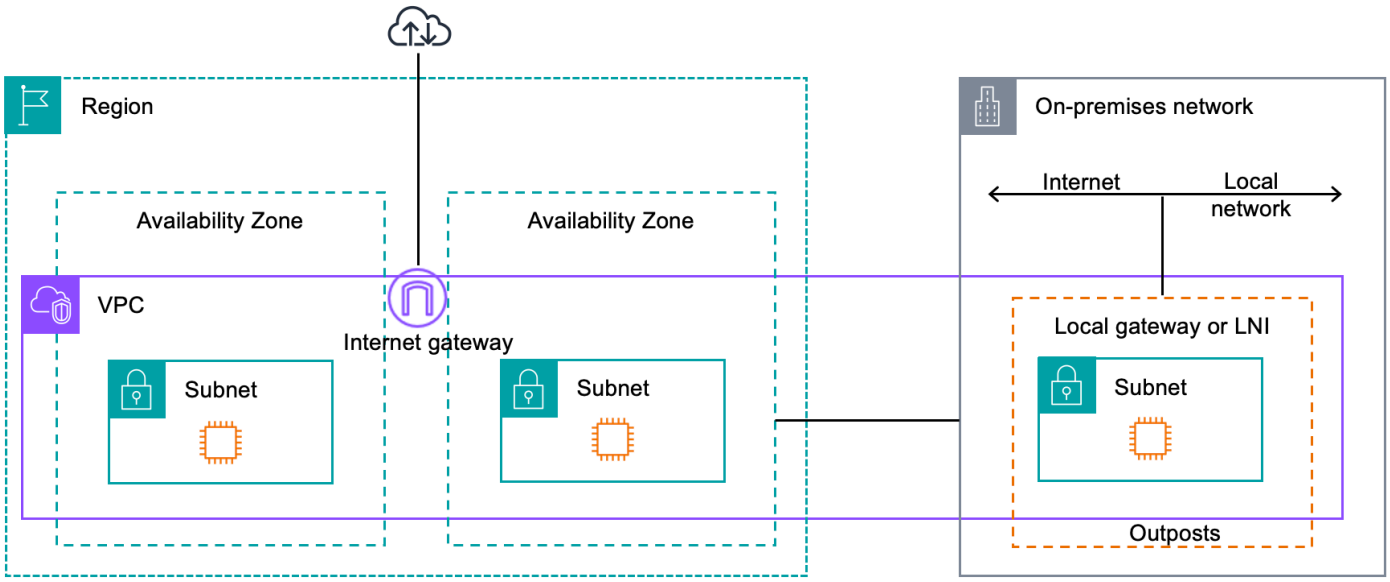
- DNS
- Service link
- Local network interfaces

Network components

AWS Outposts extends an Amazon VPC from an AWS Region to an Outpost with the VPC components that are accessible in the Region, including internet gateways, virtual private gateways, Amazon VPC Transit Gateways, and VPC endpoints. An Outpost is homed to an Availability Zone in the Region and is an extension of that Availability Zone that you can use for resiliency.

The following diagram shows the network components for your Outpost.

- An AWS Region and an on-premises network
- A VPC with multiple subnets in the Region
- An Outpost in the on-premises network
- Connectivity between the Outpost and local network provided:
  - For Outposts racks: a local gateway
  - For Outposts servers: a local network interface (LNI)



VPCs and subnets

A virtual private cloud (VPC) spans all Availability Zones in its AWS Region. You can extend any VPC in the Region to your Outpost by adding an Outpost subnet. To add an Outpost subnet to a VPC, specify the Amazon Resource Name (ARN) of the Outpost when you create the subnet.

Outposts support multiple subnets. You can specify the EC2 instance subnet when you launch the EC2 instance in your Outpost. You can't specify the underlying hardware where the instance is deployed, because the Outpost is a pool of AWS compute and storage capacity.

Each Outpost can support multiple VPCs that can have one or more Outpost subnets. For information about VPC quotas, see Amazon VPC Quotas in the *Amazon VPC User Guide*.

You create Outpost subnets from the VPC CIDR range of the VPC where you created the Outpost. You can use the Outpost address ranges for resources, such as EC2 instances that reside in the Outpost subnet.

Routing

By default, every Outpost subnet inherits the main route table from its VPC. You can create a custom route table and associate it with an Outpost subnet.

The route tables for Outpost subnets work as they do for Availability Zone subnets. You can specify IP addresses, internet gateways, local gateways, virtual private gateways, and peering connections as destinations. For example, each Outpost subnet, either through the inherited main route table, or a custom table, inherits the VPC local route. This means that all traffic in the VPC, including the Outpost subnet with a destination in the VPC CIDR remains routed in the VPC.

Outpost subnet route tables can include the following destinations:

- **VPC CIDR range** – AWS defines this at installation. This is the local route and applies to all VPC routing, including traffic between Outpost instances in the same VPC.
- **AWS Region destinations** – This includes prefix lists for Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB gateway endpoint, AWS Transit Gateways, virtual private gateways, internet gateways, and VPC peering.

  If you have a peering connection with multiple VPCs on the same Outpost, the traffic between the VPCs remains in the Outpost and does not use the service link back to the Region.

DNS

For network interfaces connected to a VPC, EC2 instances in Outposts subnets can use the Amazon Route 53 DNS Service to resolve domain names to IP addresses. Route 53 supports DNS features, such as domain registration, DNS routing, and health checks for instances running in your Outpost. Both public and private hosted Availability Zones are supported for routing traffic to specific domains. Route 53 resolvers are hosted in the AWS Region. Therefore, service link connectivity from the Outpost back to the AWS Region must be up and running for these DNS features to work.

You might encounter longer DNS resolution times with Route 53, depending on the path latency between your Outpost and the AWS Region. In such cases, you can use the DNS servers installed locally in your on-premises environment. To use your own DNS servers, you must create DHCP option sets for your on-premises DNS servers and associate them with the VPC. You must also ensure that there is IP connectivity to these DNS servers. You might also need to add routes to the local gateway routing table for reachability but this is only an option for Outposts racks with local gateway. Because DHCP option sets have a VPC scope, instances in both the Outpost subnets and the Availability Zone subnets for the VPC will try to use the specified DNS servers for DNS name resolution.

Query logging is not supported for DNS queries originating from an Outpost.

Service link

The service link is a connection from your Outpost back to your chosen AWS Region or Outposts home Region. The service link is an encrypted set of VPN connections that are used whenever the Outpost communicates with your chosen home Region. You use a virtual LAN (VLAN) to segment traffic on the service link. The service link VLAN enables communication between the Outpost and the AWS Region for both management of the Outpost and intra-VPC traffic between the AWS Region and Outpost.

Your service link is created when your Outpost is provisioned. If you have a server form factor, you create the connection. If you have a rack, AWS creates the service link. For more information, see:

-
- Application/workload routing in the *AWS Outposts High Availability Design and Architecture Considerations* AWS Whitepaper

Local network interfaces

Outposts servers include a local network interface to provide connectivity to your on-premises network. A local network interface is available only for Outposts servers running on an Outpost subnet. You can't use a local network interface from an EC2 instance on an Outposts rack or in the AWS Region. The local network interface is meant only for on-premises locations. For more information, see Local network interfaces for your Outposts servers.

Site requirements for Outposts servers

An Outpost site is the physical location where your Outpost operates. Sites are only available in select countries and territories. For more information, see AWS Outposts servers FAQs. Refer to the question: **In which countries and territories are Outposts servers available?**

This page covers the requirements for Outposts servers. For the requirements for Outposts racks, see Site requirements for Outposts racks in the *AWS Outposts User Guide for Outposts racks*.

Contents

Facility

These are the facility requirements for servers.

> **Note**
>
> Specifications are for servers under normal operating conditions. For example, acoustics may sound louder during initial installation and then operate at the rated sound power after installation is complete.

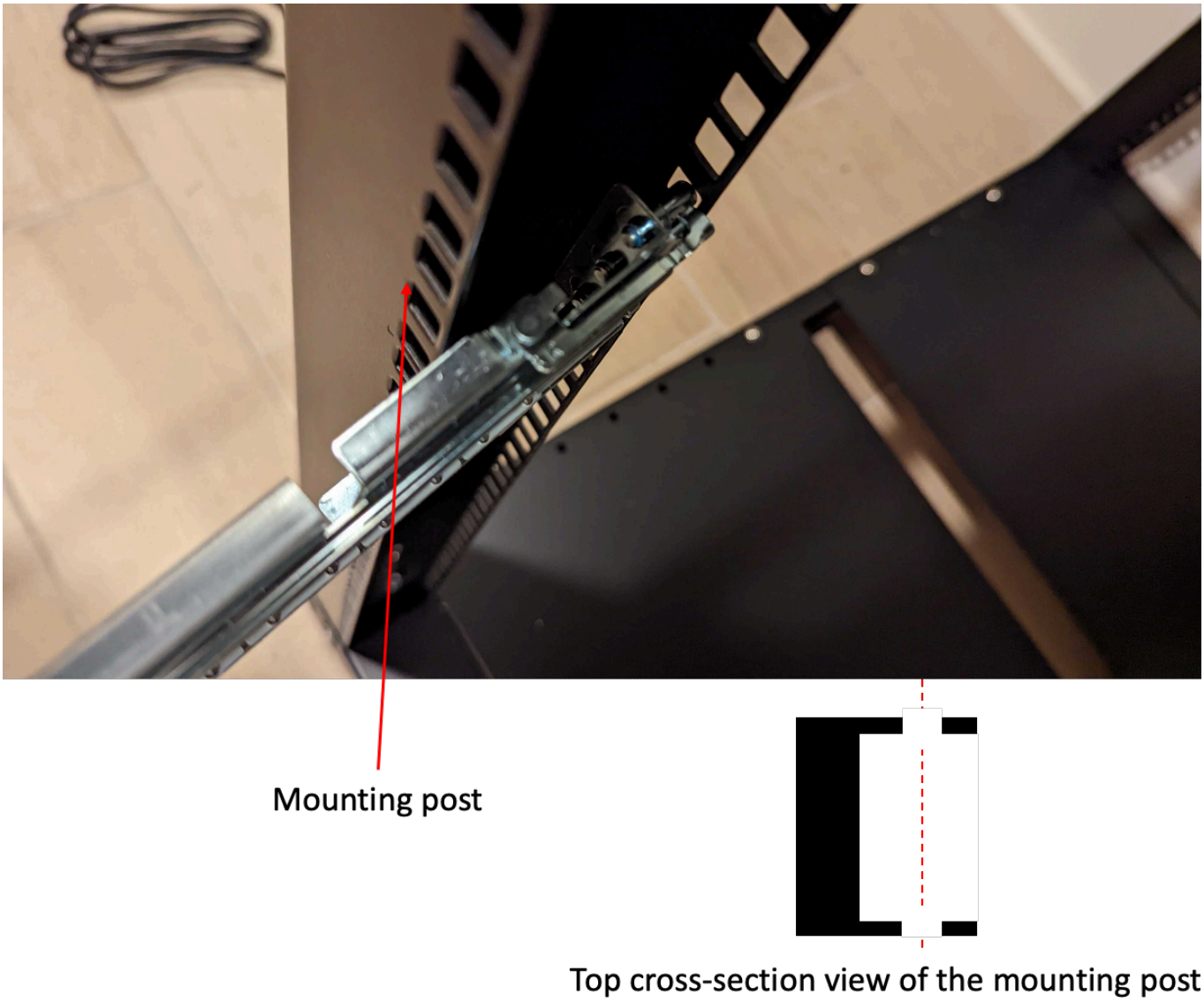
- **Temperature** – The ambient temperature must be between 41–95° F (5–35° C).

  The server will shut down when the temperature is outside this range and will restart when the temperature is back within the range.
- **Humidity** – The relative humidity must be between 8–80 percent with no condensation.
- **Air quality** – The air must be filtered using a MERV8 (or higher) filter.
- **Airflow** – The position of the server must ensure a minimum clearance of 6 inches (15 cm) between the server and walls in front of and behind the server to allow for sufficient airflow clearance.

- **Weight** – The 1U server weighs 26 pounds and the 2U server weighs 36 pounds. Confirm that the location where you intend to put the server can support the weight of the server.

  To see the weight requirements for different Outposts resources, choose **Browse catalog** in the AWS Outposts console at https://console.aws.amazon.com/outposts/.

- **Rail-kit compatibility** – The rail kit that is included in your shipping package is compatible with a standard L-shaped mounting bracket of an EIA-310-D compliant 19 inch rack. The rail kit is not compatible with a U-shaped mounting bracket, as shown in the following image.



- **Rack Placement** – We recommend the use of standard 19-inch EIA-310D racks, with a depth of at least 36 inches (914 mm). AWS provides a rail kit for rack-mounting the server.
  - Outposts 2U servers require space with the following dimensions: 3.5 inches height (88.9mm), 17.5 inches width (447 mm), 30 inches depth (762 mm)

- Outposts 1U servers require space with the following dimensions: 1.75 inches height (44.45 mm), 17.5 inches width (447 mm), 24 inches depth (610 mm)
- Mounting AWS Outposts servers vertically is not supported.
- Outposts 1U servers are the same width as Outposts 2U servers, but half the height and less depth

If you do not place the server in a rack, you must still meet the other site requirements.

- **Serviceability** – Outposts servers are front-aisle serviceable.
- **Acoustics** – rated to be less than 78 dBA sound power at temperatures of 80 ° F (27 ° C) and meets GR-63 CORE NEBS compliance.
- **Seismic bracing** – To the extent required by regulation or code, you will install and maintain appropriate seismic anchorage and bracing for the server while it is in your facility.
- **Elevation** – The elevation of the room where the rack is installed must be below 10,005 feet (3,050 meters).
- **Cleaning** – Wipe surfaces with damp wipes that contain approved antistatic cleaning chemicals.

Networking

Each Outposts server includes non-redundant physical uplink ports. Ports have their own speed and connector requirements as detailed below.

Port label	Speed	Connector on the upstream networkin g device	Traffic
Port 3	10Gbe	SFP+	Both service and LNI link traffic – QSFP+ breakout cable (10 feet/3 m) segments traffic.

Service link firewall

UDP and TCP 443 must be statefully listed in the firewall.

Protocol	Source Port	Source Address	Destination Port	Destination Address
UDP	1024-65535	Service Link IP	53	DNS server
UDP	443, 1024-65535	Service Link IP	443	Outposts Service Link endpoints
TCP	1024-65535	Service Link IP	443	Outposts Registration endpoints

You can use an AWS Direct Connect connection or a public internet connection to connect the Outpost back to the AWS Region. For Outposts service link connectivity, you can use NAT or PAT at your firewall or edge router. Service link establishment is always initiated from the Outpost.

Service link maximum transmission unit (MTU)

The network must support 1500-bytes MTU between the Outpost and the service link endpoints in the parent AWS Region. For more information about the service link, see AWS Outposts connectivity to AWS Regions in the *AWS Outposts user guide for servers*.

Service link bandwidth recommendations

For an optimal experience and resiliency, AWS requires that you use redundant connectivity of at least 500 Mbps and a maximum of 175 ms round trip latency for the service link connection to the AWS Region. The maximum utilization for each Outposts server is 500 Mbps. To increase the connection speed, use multiple Outposts servers. For example, if you have three AWS Outposts servers, the maximum connection speed increases to 1.5 Gbps (1,500 Mbps). For more information, see Service link traffic for servers in the *AWS Outposts user guide for servers*.

Your AWS Outposts service link bandwidth requirements vary depending on workload characteristics, such as AMI size, application elasticity, burst speed needs, and Amazon VPC traffic to the Region. Note that AWS Outposts servers do not cache AMIs. AMIs are downloaded from the Region with every instance launch.

To receive a custom recommendation about the service link bandwidth required for your needs, contact your AWS sales representative or APN partner.

Power

These are the power requirements for Outposts servers.

Requirements

- Power support
- Power draw
- Power cable
- Power redundancy

Power support

Servers are rated up to 1600W 90-264 VaC 47/63 Hz AC power.

Power draw

To see the power draw requirements for different Outposts resources, choose **Browse catalog** in the AWS Outposts console at https://console.aws.amazon.com/outposts/.

Power cable

The server ships with an IEC C14-C13 power cable.

Power cabling from server to rack

Use the provided IEC C14-C13 power cable to connect the server to the rack.

Power cabling from server to wall outlet

To connect the server to a standard wall outlet, you must use either an adapter for the C14 inlet or a country-specific power cord.

Ensure that you have the correct adapter or power cord for your region to save time during server installation.

- In the United States, you need an IEC C13 to NEMA 5-15P power cord.
- In parts of Europe, you might need an IEC C13 to CEE 7/7 power cord.
- In India, you need an IEC C13 to IS1293 power cord.

Power redundancy

Servers include multiple power connections and ship with cables to enable power redundant operation. We recommend power redundancy, but redundancy is not required.

Servers do not include an Uninterruptible Power Supply (UPS).

Order fulfillment

To fulfill the order, AWS will ship the Outposts server equipment, including rail mounts and required power and network cables, to the address that you provided. The box that the server is shipped in has the following dimensions:

- Box with a 2U server:
  - Length: 44 inches / 111.8 cm
  - Height: 26.5 inches / 67.3 cm
  - Width: 17 inches / 43.2 cm
- Box with a 1U server:
  - Length: 34.5 inches / 87.6 cm
  - Height: 24 inches / 61 cm
  - Width: 9 inches / 22.9 cm

Your team or a third-party provider must install the equipment. For more information, see Service link traffic for servers in the *AWS Outposts user guide for servers*.

The installation is complete when you confirm that the Amazon EC2 capacity for your Outposts server is available from your AWS account.

Get started with Outposts servers

Order an Outposts server to get started. After installation of your Outpost equipment, launch an Amazon EC2 instance and configure connectivity to your on-premises network.

Tasks

- Create an Outpost and order Outpost capacity
- Launch an instance on your Outposts server

Create an Outpost and order Outpost capacity

To begin using AWS Outposts, log in with your AWS account. Create a site and an Outpost. Then, place an order for the Outposts servers that you require.

Prerequisites

- Review the available configurations for your Outposts servers.
- An Outpost site is the physical location for your Outpost equipment. Before ordering capacity, verify that your site meets the requirements. For more information, see Site requirements for Outposts servers.
- You must have an AWS Enterprise Support plan or an AWS Enterprise On-Ramp Support plan.
- Determine which AWS account you will use to create the Outposts site, create the Outpost, and place the order. Monitor the email associated with this account for information from AWS.

Tasks

- Step 1: Create a site
- Step 2: Create an Outpost
- Step 3: Place the order
- Step 4: Modify instance capacity
- Next steps

Step 1: Create a site

Create a site to specify the operating address. The operating address is the location where you will install and run your Outposts servers. After you create the site, AWS Outposts assigns an ID to your site. You must specify this site when you create an Outpost.

Prerequisites

- Determine the operating address.

To create a site

1. Sign in to AWS.
2. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
3. To select the parent AWS Region, use the Region selector in the upper-right corner of the page.
4. In the navigation pane, choose **Sites**.
5. Choose **Create site**.
6. For **Supported hardware type**, choose **Servers only**.
7. Enter the name, description, and operating address for your site.
8. (Optional) For **Site notes**, enter any other information that might be useful for AWS to know about the site.
9. Choose **Create site**.

Step 2: Create an Outpost

Create an Outpost for each server. An Outpost can only be associated with a single server. You'll specify this Outpost when you place the order.

Prerequisites

- Determine the AWS Availability Zone to associate with your site.

To create an Outpost

1. In the navigation pane, choose **Outposts**.

2. Choose **Create Outpost**.
3. Choose **Servers**.
4. Enter the name and a description for your Outpost.
5. Choose an Availability Zone for your Outpost.
6. For **Site ID**, choose your site.
7. Choose **Create Outpost**.

Step 3: Place the order

Place an order for the Outposts servers that you need.

> **Important**
>
> You can't edit an order after you submit it so review all details carefully before submission. If you need to change an order, contact AWS Support Center.

Prerequisites

- Determine how you will pay for the order. You can pay all upfront, partially upfront, or nothing upfront. If you choose the partial-upfront or no-upfront payment option, you'll pay monthly charges over the term.

  The pricing includes delivery, infrastructure service maintenance, and software patches and upgrades.
- Determine whether the shipping address is different from the operating address that you specified for the site.

To place an order

1. In the navigation pane, choose **Orders**.
2. Choose **Place order**.
3. For **Supported hardware type**, choose **Servers**.
4. To add capacity, choose a configuration.
5. Choose **Next**.

6. Choose **Use an existing Outpost** and select your Outpost.
7. Choose **Next**.
8. Select a contract term and payment option.
9. Specify the shipping address. You can specify a new address or select the site's operating address. If you select the operating address, be aware that any future change to the site's operating address will not propagate to existing orders. If you need to change the shipping address on an existing order, contact your AWS Account Manager.
10. Choose **Next**.
11. On the **Review and order** page, verify that your information is correct and edit as needed. You will not be able to edit the order after you submit it.
12. Choose **Place order**.

Step 4: Modify instance capacity

The capacity of each new Outpost order is configured with a default capacity configuration. You can convert the default configuration to create various instances to meet your business needs. To do so, you create a capacity task, specify the instance sizes and quantity, and run the capacity task to implement the changes.

> **Note**
>
> - You can change the quantity of instance sizes after you place the order for your Outposts.
> - Instances sizes and quantities are defined at the Outpost level.
> - Instances are placed automatically based on best practices.

To modify instance capacity

1. From the AWS Outposts console'sAWS Outposts left navigation pane, choose **Capacity tasks**.
2. On the **Capacity tasks** page, choose **Create capacity task**.
3. On the **Getting started** page, choose the order.
4. To modify capacity, you can use the steps in the console or upload a JSON file.

Console steps

1. Choose **Modify a new Outpost capacity configuration**.
2. Choose **Next**.
3. On the **Configure instance capacity** page, each instance type shows one instance size with the maximum quantity preselected. To add more instance sizes, choose **Add instance size**.
4. Specify the instance quantity and note the capacity that is displayed for that instance size.
5. View the message at the end of each instance-type section that informs you if you are over or under capacity. Make adjustments at the instance size or quantity level to optimize your total available capacity.
6. You can also request AWS Outposts to optimize the instance quantity for a specific instance size. To do so:

   a. Choose the instance size.

   b. Choose **Auto-balance** at the end of the related instance-type section.
7. For each instance type, ensure that the instance quantity is specified for at least one instance size.
8. Choose **Next**.
9. On the **Review and create** page, verify the updates that you are requesting.
10. Choose **Create**. AWS Outposts creates a capacity task.
11. On the capacity task page, monitor the status of the task.

> **Note**
>
> AWS Outposts might request you to stop one or more running instances to enable running the capacity task. After you stop these instances, AWS Outposts will run the task.

Upload JSON file

1. Choose **Upload a capacity configuration**.
2. Choose **Next**.
3. On the **Upload capacity configuration plan** page, upload the JSON file that specifies the instance type, size, and quantity.

Example

Example JSON file:

```
{
    "RequestedInstancePools": [
        {
            "InstanceType": "c5.24xlarge",
            "Count": 1
        },
        {
            "InstanceType": "m5.24xlarge",
            "Count": 2
        }
    ]
}
```

4. Review the contents of the JSON file in the **Capacity configuration plan** section.
5. Choose **Next**.
6. On the **Review and create** page, verify the updates that you are requesting.
7. Choose **Create**. AWS Outposts creates a capacity task.
8. On the capacity task page, monitor the status of the task.

> **Note**
>
> AWS Outposts might request you to stop one or more running instances to enable running the capacity task. After you stop these instances, AWS Outposts will run the task.

Next steps

You can view the status of your order using the AWS Outposts console. The initial status of your order is **Order received**. If you have any questions about your order, contact AWS Support Center.

To fulfill the order, AWS will schedule a delivery date.

You are responsible for all installation tasks, including physical installation and network configuration. You can contract a third-party to perform these tasks for you. Whether you do

the installation or contract to a third-party, installation requires IAM credentials in the AWS account that contains the Outpost to verify the identity of the new device. You are responsible for providing and managing this access. For more information, see the Server installation guide.

The installation is complete when Amazon EC2 capacity for your Outpost is available from your AWS account. After the capacity is available, you can launch Amazon EC2 instances on your Outposts server. For more information, see the section called “Launch an instance”.

Launch an instance on your Outposts server

After your Outpost is installed and the compute and storage capacity is available for use, you can get started by creating resources. For example, you can launch Amazon EC2 instances.

Prerequisite

You must have an Outpost installed at your site. For more information, see Create an Outpost and order Outpost capacity.

Tasks

- Step 1: Create a subnet
- Step 2: Launch an instance on the Outpost
- Step 3: Configure connectivity
- Step 4: Test the connectivity

Step 1: Create a subnet

You can add Outpost subnets to any VPC in the AWS Region for the Outpost. When you do so, the VPC also spans the Outpost. For more information, see Network components.

> **Note**
>
> If you are launching an instance in an Outpost subnet that has been shared with you by another AWS account, skip to Step 2: Launch an instance on the Outpost.

To create an outpost subnet

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.

2. On the navigation pane, choose **Outposts**.
3. Select the Outpost, and then choose **Actions**, **Create subnet**. You are redirected to create a subnet in the Amazon VPC console. We select the Outpost for you and the Availability Zone that the Outpost is homed to.
4. Select a VPC and specify an IP address range for the subnet.
5. Choose **Create**.
6. After the subnet is created, you must enable the subnet for local network interfaces. Use the modify-subnet-attribute command from the AWS CLI. You must specify the position of the network interface on the device index. All instances launched in an enabled Outpost subnet use this device position for local network interfaces. The following example uses a value of 1 to specify a secondary network interface.

```
aws ec2 modify-subnet-attribute \
    --subnet-id subnet-1a2b3c4d \
    --enable-lni-at-device-index 1
```

Step 2: Launch an instance on the Outpost

You can launch EC2 instances in the Outpost subnet that you created, or in an Outpost subnet that has been shared with you. Security groups control inbound and outbound VPC traffic for instances in an Outpost subnet, just as they do for instances in an Availability Zone subnet. To connect to an EC2 instance in an Outpost subnet, you can specify a key pair when you launch the instance, just as you do for instances in an Availability Zone subnet.

Considerations

- Instances on Outposts servers include instance store volumes but not EBS volumes. Choose an instance size with enough instance storage to meet the needs of your application. For more information, see Instance store volumes and Create an instance store-backed AMI in the *Amazon EC2 User Guide*.
- You must use an Amazon EBS-backed AMI with only a single EBS snapshot. AMIs with more than one EBS snapshot are not supported.
- The data on instance store volumes persists after an instance reboot but does not persist after instance termination. To retain the long-term data on your instance store volumes beyond the lifetime of the instance, be sure to back up the data to persistent storage, such as an Amazon S3 bucket or a network storage device in your on-premises network.

- To use block data or boot volumes backed by compatible third-party storage, you must provision and configure these volumes for use with EC2 instances on Outposts. For more information, see *Third-party block storage*.
- To connect an instance in an Outpost subnet to your on-premises network, you must add a local network interface, as described in the following procedure.

To launch instances in your Outpost subnet

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Outposts**.
3. Select the Outpost, and then choose **Actions, View details**.
4. On the **Outpost summary** page, choose **Launch instance**. You are redirected to the instance launch wizard in the Amazon EC2 console. We select the Outpost subnet for you, and show you only the instance types that are supported by your Outposts servers.
5. Choose an instance type that is supported by your Outposts servers. Note that instances that appear grayed out are not available.
6. (Optional) You can add a local network interface now or after you create the instance. To add it now, expand **Advanced network configuration** and choose **Add network interface**. Choose the Outpost subnet. This creates a network interface for the instance using device index 1. If you specified 1 as the local network interface device index for the Outpost subnet, this network interface is the local network interface for the instance. Alternatively, to add it later, see Add a local network interface.
7. (Optional) You can add a third-party data volume.
    a. Expand **Configure storage**. Next to **External storage volume**, choose **Edit**.
    b. For **Storage Network Protocol**, choose **iSCSI**.
    c. Enter the Initiator IQN, then add the target IP address, the port, and the IQN of the external storage array.
8. Complete the wizard to launch the instance in your Outpost subnet. For more information, see Launch an EC2 instance in the *Amazon EC2 User Guide*:

Step 3: Configure connectivity

If you did not add a local network interface to your instance during instance launch, you must do so now. For more information, see Add a local network interface.

You must configure the local network interface for the instance with an IP address from your local network. For information, see the documentation for the operating system running on the instance. Search for information about configuring additional network interfaces and secondary IP addresses.

Step 4: Test the connectivity

You can test connectivity by using the appropriate use cases.

Test connectivity from your local network to the Outpost

From a computer in your local network, run the `ping` command to the Outpost instance's local network interface IP address.

```
ping 10.0.3.128
```

The following is example output.

```
Pinging 10.0.3.128

Reply from 10.0.3.128:  bytes=32 time=<1ms TTL=128
Reply from 10.0.3.128:  bytes=32 time=<1ms TTL=128
Reply from 10.0.3.128:  bytes=32 time=<1ms TTL=128

Ping statistics for 10.0.3.128
Packets:  Sent = 3,  Received = 3,  Lost = 0 (0% lost)

Approximate round trip time in milliseconds
Minimum = 0ms,  Maximum = 0ms,  Average = 0ms
```

Test the connectivity from an Outpost instance to your local network

Depending on your operating system, use **ssh** or **rdp** to connect to the private IP address of your Outpost instance. For information about connecting to an EC2 instance, see Connect to your EC2 instance in the *Amazon EC2 User Guide*.

After the instance is running, run the `ping` command to an IP address of a computer in your local network. In the following example, the IP address is 172.16.0.130.

```
ping 172.16.0.130
```

The following is example output.

```
Pinging 172.16.0.130

Reply from 172.16.0.130:  bytes=32 time=<1ms TTL=128
Reply from 172.16.0.130:  bytes=32 time=<1ms TTL=128
Reply from 172.16.0.130:  bytes=32 time=<1ms TTL=128

Ping statistics for 172.16.0.130
Packets:  Sent = 3,  Received = 3,  Lost = 0 (0% lost)

Approximate round trip time in milliseconds
Minimum = 0ms,  Maximum = 0ms,  Average = 0ms
```

Test connectivity between the AWS Region and the Outpost

Launch an instance in the subnet in the AWS Region. For example, use the run-instances command.

```
aws ec2 run-instances \
    --image-id ami-abcdefghi1234567898 \
    --instance-type c5.large \
    --key-name MyKeyPair \
    --security-group-ids sg-1a2b3c4d123456787 \
    --subnet-id subnet-6e7f829e123445678
```

After the instance is running, perform the following operations:

1. Get the private IP address of the instance in the AWS Region. This information is available in the Amazon EC2 console on the instance detail page.
2. Depending on your operating system, use **ssh** or **rdp** to connect to the private IP address of your Outpost instance.
3. Run the **ping** command from your Outpost instance, specifying the IP address of the instance in the AWS Region.

   ```
   ping 10.0.1.5
   ```

   The following is example output.

   ```
   Pinging 10.0.1.5
   ```

```
Reply from 10.0.1.5:  bytes=32 time=<1ms TTL=128
Reply from 10.0.1.5:  bytes=32 time=<1ms TTL=128
Reply from 10.0.1.5:  bytes=32 time=<1ms TTL=128

Ping statistics for 10.0.1.5
Packets:  Sent = 3,  Received = 3,  Lost = 0 (0% lost)

Approximate round trip time in milliseconds
Minimum = 0ms,  Maximum = 0ms,  Average = 0ms
```

AWS Outposts connectivity to AWS Regions

AWS Outposts supports wide area network (WAN) connectivity through the service link connection.

> **Note**
>
> You can't use private connectivity for your service link connection that connects your Outposts server to your AWS Region or AWS Outposts home Region.

Contents

Connectivity through service link

During AWS Outposts provisioning, you or AWS creates a service link connection that connects your Outposts server to your chosen AWS Region or home Region. The service link is an encrypted set of VPN connections that are used whenever the Outpost communicates with your chosen home Region. You use a virtual LAN (VLAN) to segment traffic on the service link. The service link VLAN enables communication between the Outpost and the AWS Region for both management of the Outpost and intra-VPC traffic between the AWS Region and Outpost.

The Outpost is able to create the service link VPN back to the AWS Region through public Region connectivity. To do so, the Outpost needs connectivity to the AWS Region's public IP ranges, either through the public internet or AWS Direct Connect public virtual interface. This connectivity can be through specific routes in the service link VLAN, or through a default route of 0.0.0.0/0. For more information about the public ranges for AWS, see AWS IP address ranges in the *Amazon VPC User Guide*.

After the service link is established, the Outpost is in service and managed by AWS. The service link is used for the following traffic:

- Management traffic to the Outpost through the service link, including internal control plane traffic, internal resource monitoring, and updates to firmware and software.

- Traffic between the Outpost and any associated VPCs, including customer data plane traffic.

Service link maximum transmission unit (MTU) requirements

The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection.

Note the following:

- The network **must** support 1500-bytes MTU between the Outpost and the service link endpoints in the parent AWS Region.
- The traffic that goes from an instance in Outposts to an instance in the Region has an MTU of 1300 bytes, which is lower than the required MTU of 1500 bytes due to packet overheads.

Service link bandwidth recommendations

For an optimal experience and resiliency, AWS requires that you use redundant connectivity of at least 500 Mbps and a maximum of 175 ms round trip latency for the service link connection to the AWS Region. The maximum utilization for each Outposts server is 500 Mbps. To increase the connection speed, use multiple Outposts servers. For example, if you have three AWS Outposts servers, the maximum connection speed increases to 1.5 Gbps (1,500 Mbps). For more information, see Service link traffic for servers.

Your AWS Outposts service link bandwidth requirements vary depending on workload characteristics, such as AMI size, application elasticity, burst speed needs, and Amazon VPC traffic to the Region. Note that AWS Outposts servers do not cache AMIs. AMIs are downloaded from the Region with every instance launch.

To receive a custom recommendation about the service link bandwidth required for your needs, contact your AWS sales representative or APN partner.

Redundant internet connections

When you build connectivity from your Outpost to the AWS Region, we recommend that you create multiple connections for higher availability and resiliency. For more information, see AWS Direct Connect Resiliency Recommendations.

If you need connectivity to the public internet, you can use redundant internet connections and diverse internet providers, just as you would with your existing on-premises workloads.

Updates and the service link

AWS maintains a secure network connection between your Outposts server and its parent AWS Region. This network connection, called the service link, is essential in managing the Outpost by providing intra-VPC traffic between the Outpost and AWS Region. AWS Well-Architected best practices recommend deploying applications across two Outposts parented to different Availability Zones with an active-active design. For more information, see AWS Outposts High Availability Design and Architecture Considerations.

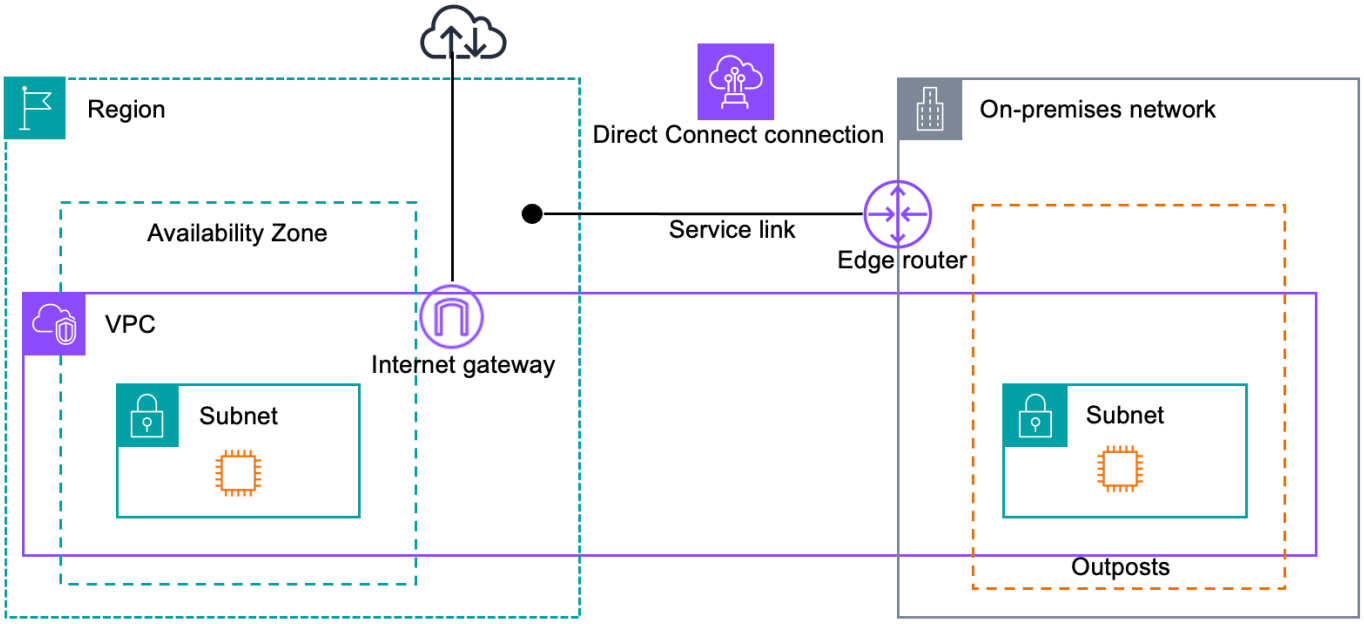
The service link is regularly updated to maintain operational quality and performance. During maintenance, you might observe brief periods of latency and packet loss on this network resulting in impact on workloads that are dependent on VPC connectivity to resources hosted in-region. However, traffic traversing the Local Network Interfaces (LNI) will not be impacted. You can avoid impact to your application by following AWS Well-Architected best practices and by ensuring your applications are resilient to failures or maintenance activities affecting a single Outposts server.

Firewalls and the service link

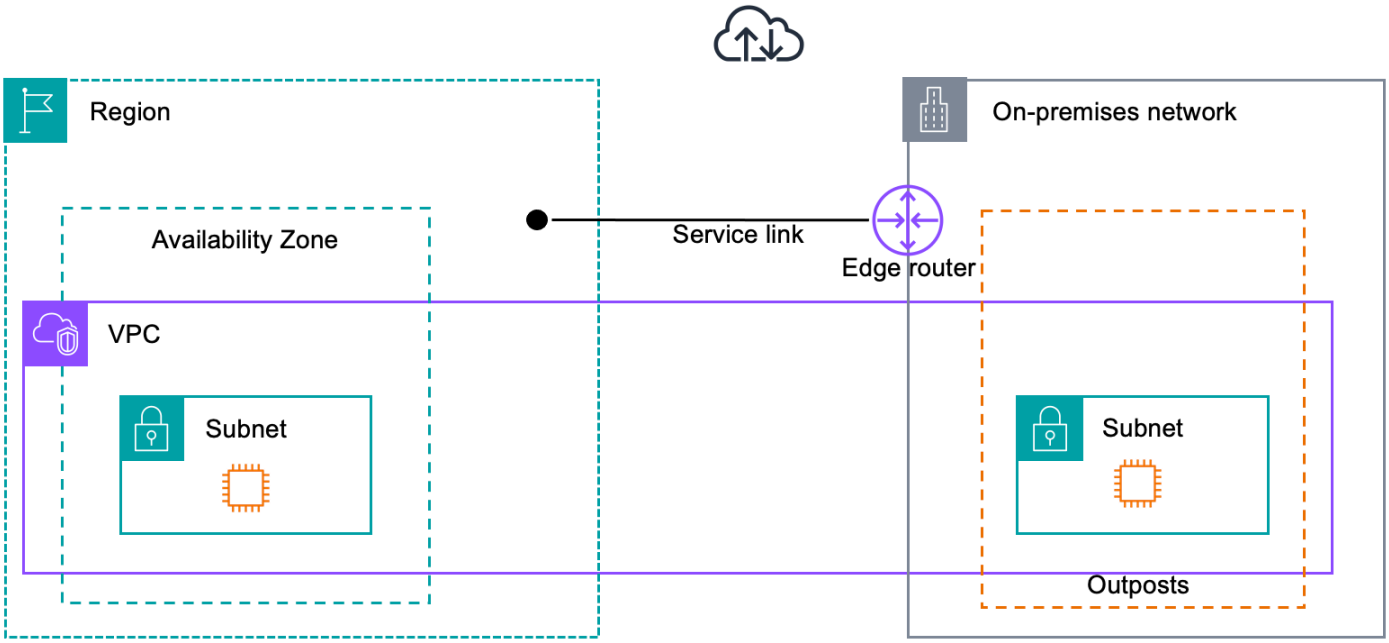
This section discusses firewall configurations and the service link connection.

In the following diagram, the configuration extends the Amazon VPC from the AWS Region to the Outpost. An AWS Direct Connect public virtual interface is the service link connection. The following traffic goes over the service link and the AWS Direct Connect connection:

- Management traffic to the Outpost through the service link
- Traffic between the Outpost and any associated VPCs



If you are using a stateful firewall with your internet connection to limit connectivity from the public internet to the service link VLAN, you can block all inbound connections that initiate from the internet. This is because the service link VPN initiates only from the Outpost to the Region, not from the Region to the Outpost.



If you use a firewall to limit the connectivity from the service link VLAN, you can block all inbound connections. You must allow outbound connections back to the Outpost from the AWS Region as

per the following table. If the firewall is stateful, outbound connections from the Outpost that are allowed, meaning that they were initiated from the Outpost, should be allowed back inbound.

Protocol	Source Port	Source Address	Destination Port	Destination Address
UDP	1024-65535	Service Link IP	53	DNS server
UDP	443, 1024-65535	Service Link IP	443	AWS Outposts Service Link endpoints
TCP	1024-65535	Service Link IP	443	AWS Outposts Registration endpoints

Note

Instances in an Outpost can't use the service link to communicate with instances in another Outposts. Leverage routing through the local gateway or local network interface to communicate between Outposts.

Outposts server network troubleshooting

Use this checklist to help troubleshoot a service link that has a status of `DOWN`.

Initial assessment

Verify the status of the service link through Amazon CloudWatch metrics:

1. Monitor the **ConnectedStatus** metric in the AWS Outposts namespace.
2. If the average value is less than 1, this confirms that the service link is impaired.
3. If the service link is impaired, complete the steps in the following sections to resolve and reestablish the connection.

Step 1. Check physical connectivity

1. Verify you are using the provided QSFP breakout cable. If issues persist, test with a different QSFP breakout cable if available.
2. Verify that the QSFP breakout cable in the Outposts server is firmly seated.
3. Verify that **cable 1** (LNI) is firmly seated in the switch.
4. Verify that **cable 2** (service link) is firmly seated in the switch.
5. Complete a general switch-sanity check such as, checking link lights.

Step 2. Test the Outposts server connection to AWS

Create a serial connection to the Outposts server and perform the following tests:

1. Test the links.
   a. If successful, proceed with the next test.
   b. If it fails, Verify network configuration.
2. Test for DNS resolution.
   a. If successful, proceed with the next test.
   b. If it fails, Check firewall rules.
3. Test for access to the AWS Region.
   a. If successful, proceed to reestablish the connection.
   b. If it fails, Verify MTU.

Verify network configuration

Ensure that your switch meets the following specifications:

- **Basic configuration** — The service link port must be an untagged access port to a VLAN with a gateway and a route to AWS endpoints.
- **Link speed** — The switch port must have link speed set to 10 Gb and auto-negotiation must be turned off.

Verify MTU

The network must support 1500-bytes MTU between the Outpost and the service link endpoints in the parent AWS Region. For more information about the service link, see AWS Outposts connectivity to AWS Regions.

Check firewall rules

If you use a firewall to limit the connectivity from the service link VLAN, you can block all inbound connections. You must allow outbound connections back to the Outpost from the AWS Region as per the following table. If the firewall is stateful, outbound connections from the Outpost that are allowed, meaning that they were initiated from the Outpost, should be allowed back inbound.

Protocol	Source Port	Source Address	Destination Port	Destination Address
UDP	1024-65535	Service Link IP	53	DNS server
UDP	443, 1024-65535	Service Link IP	443	AWS Outposts Service Link endpoints
TCP	1024-65535	Service Link IP	443	AWS Outposts Registration endpoints

Step 3. Reestablish connectivity

If the previous checks pass but the service link remains `DOWN` (**ConnectedStatus** is less than 1 in CloudWatch), then follow the steps in Authorize the Outposts server using the Outpost Configuration Tool to reestablish the connection.

> **Note**
>
> If the service link remains down, create a case at the AWS Support Center.

Return an Outposts server

If AWS Outposts detects a defect in the server, we will inform you, start the replacement process to send you a new server, and provide you with the return label through the AWS Outposts console. You will not be charged a shipping fee when you return an Outposts server. However, if you return a server that is damaged, you might incur a cost.

To get started, complete the following steps.

Tasks

Step 1: Prepare the server for return

To prepare the server for return, unshare resources, backup data, delete local network interfaces and terminate active instances.

1. If the Outpost's resources are shared, you must unshare these resources.

   You can unshare a shared Outpost resource in one of the following ways:

   - Use the AWS RAM console. For more information, see Updating a resource share in the *AWS RAM User Guide*.
   - Use the AWS CLI to run the disassociate-resource-share command.

   For the list of Outpost resources that can be shared, see Shareable Outpost resources.
2. Create backups of the data stored in the instance storage of the Amazon EC2 instances running on the AWS Outposts server.
3. Delete the local network interfaces associated with the instances that were running on the server.
4. Terminate the active instances associated with subnets on your Outpost. To terminate the instances, follow the instructions in Terminate your instance in the *Amazon EC2 User Guide*.

5. Destroy the Nitro Security Key (NSK) to cryptographically shred your data on the server. To destroy the NSK, follow the instructions in Cryptographically shred server data.

Step 2: Print the return label

> **Important**
>
> You must only use the return label that AWS provides because it contains specific information, such as the Asset ID, about the server that you are returning. Do not create your own return label.

To obtain your return label:

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Orders**.
3. Under **Replacement order summary**, choose **Print return label** and choose the configuration ID of the server that you plan to return.

> **Note**
>
> Returning your Outposts servers before the current subscription ends will not terminate any outstanding charges associated with this Outpost.

Step 3: Pack the server

To pack your server, use the box and packaging material provided by AWS.

1. Pack the server in one of the following boxes:

   - The box and packaging material that the server originally came in.
   - The box and packaging material that the replacement server came in.

   Alternatively, contact AWS Support Center to request a box.
2. Affix the return label that AWS provided, to the outside of the box.

> **Important**
>
> Verify that the Asset ID on the return label matches the Asset ID on the server that you are returning.
> The Asset ID is located on the pull-out tab on the front of the server. Example: `1203779889` or `9305589922`

3. Seal the box securely.

Step 4: Return the server through the courier

You must return the server through the designated courier for your country. You can deliver the server to the courier or schedule the day and time that you prefer for the courier to pick up the server. The return label that AWS provides contains the correct address to return the server.

The following table shows who to contact for the country you are shipping from:

Country	Contact
Argentina	Contact AWS Support Center. In your request, include the following information: • The tracking number that is on the AWS-provided return label • The date and time that you prefer the courier to pick up the server • A contact name • A phone number • An email address
Bahrain	
Brazil	
Brunei	
Canada	
Chile	
Colombia	
Hong Kong	
India	
Indonesia	

Country	Contact
Japan	
Malaysia	
Nigeria	
Oman	
Panama	
Peru	
Philippines	
Serbia	
Singapore	
South Africa	
South Korea	
Taiwan	
Thailand	
United Arab Emirates	
Vietnam	
Mexico	AWS contacts DB Schenker and requests a pickup from your location. DB Schenker then contacts you to schedule the date and time for the pickup.

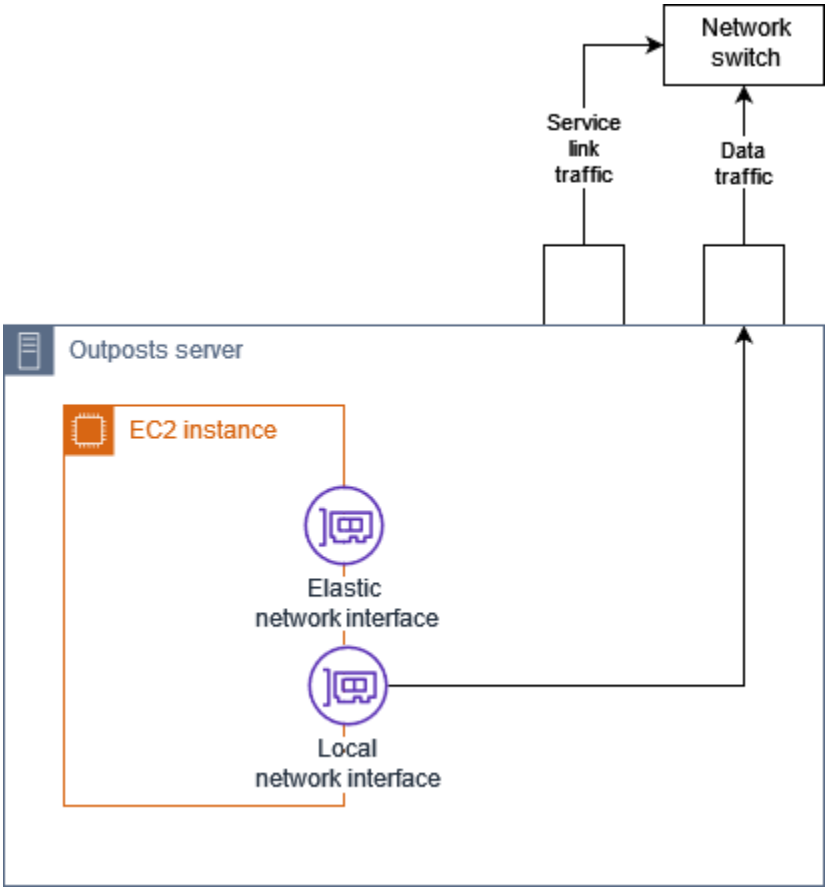
Country	Contact
United States of America	Contact UPS. You can return the server in the following ways: • Return the server during a routine UPS pickup at your site. • Drop-off the server at a UPS location. • Schedule a pickup for a date and time you prefer. Enter the tracking number from the AWS-provided return label for free shipping.
All other countries	Contact DHL. You can return the server in the following ways: • Drop-off the server at a DHL location. • Schedule a pickup for a date and time you prefer. Enter the DHL Waybill number from the AWS-provided return label for free shipping. If you get the following error `Courier pickup can't be scheduled for an import shipment`, it usually means that the pickup country that you selected does not match the pickup country on the return shipment label. Select the country where the shipment originates from and try again.

Local network interfaces for your Outposts servers

With Outposts servers, a *local network interface* is a logical networking component that connects the Amazon EC2 instances in your Outposts subnet to your on-premises network.

A local network interface runs directly on your local area network. With this type of local connectivity, you don't need routers or gateways to communicate with your on-premises equipment. Local network interfaces are named similarly to network interfaces or elastic network interfaces. We distinguish between the two interfaces by always using *local* when we refer to local network interfaces.

After you enable local network interfaces on an Outpost subnet, you can configure the EC2 instances in the Outpost subnet to include a local network interface in addition to the elastic network interface. The local network interface connects to the on-premises network while the network interface connects to the VPC. The following diagram shows an EC2 instance on an Outposts server with both an elastic network interface and a local network interface.



You must configure the operating system to enable the local network interface to communicate on your local area network, just as you would for any other on-premises equipment. You can't use DHCP option sets in a VPC to configure a local network interface because a local network interface runs on your local area network.

The elastic network interface works exactly as it does for instances in an Availability Zone subnet. For example, you can use the VPC network connection to access the public Regional endpoints for AWS services, or you can use interface VPC endpoints to access AWS services using AWS PrivateLink. For more information, see AWS Outposts connectivity to AWS Regions.

Contents

Local network interface basics

Local network interfaces provide access to a physical layer-two network. A VPC is a virtualized layer-three network. Local network interfaces do not support VPC networking components. These components include security groups, network access control lists, virtualized routers or route tables, and flow logs. The local network interface does not provide the Outposts server with visibility into VPC layer-three flows. The host operating system of the instance does have full visibility into frames from the physical network. You can apply standard firewall logic to information within these frames. However, this communication happens inside the instance but outside the purview of the virtualized constructs.

Considerations

- Local network interfaces support ARP and DHCP protocols. They do not support general L2 broadcast messages.
- Quotas for local network interfaces comes out of your quota for network interfaces. For more information, see Network interface quotas in the *Amazon VPC User Guide*.
- Each EC2 instance can have one local network interface.
- A local network interface can't use the primary network interface of the instance.
- Outposts servers can host multiple EC2 instances, each with a local network interface.

> **Note**
>
> EC2 instances within the same server can communicate directly without sending data outside the Outposts server. This communication includes traffic over a local network interface or elastic network interfaces.

- Local network interfaces are available only for instances running in an Outposts subnet on an Outposts server.
- Local network interfaces do not support promiscuous mode or MAC address spoofing.

Performance

The local network interface of each instance size provides a portion of the physical 10 GbE available bandwidth. The following table lists the network performance for each instance type:

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
c6id.large	0.15625	2.5
c6id.xlarge	0.3125	2.5
c6id.2xlarge	0.625	2.5
c6id.4xlarge	1.25	2.5
c6id.8xlarge	2.5	2.5
c6id.12xlarge	3.75	3.75
c6id.16xlarge	5	5
c6id.24xlarge	7.5	7.5
c6id.32xlarge	10	10
c6gd.medium	0.15625	4
c6gd.large	0.3125	4

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
c6gd.xlarge	0.625	4
c6gd.2xlarge	1.25	4
c6gd.4xlarge	2.5	4
c6gd.8xlarge	4.8	4.8
c6gd.12xlarge	7.5	7.5
c6gd.16xlarge	10	10

Security groups

By design, the local network interface does not use security groups in your VPC. A security group controls inbound and outbound *VPC traffic*. The local network interface is not attached to the VPC. The local network interface is attached to your local network. To control inbound and outbound traffic on the local network interface, use a firewall or similar strategy, just as you would with the rest of your on-premises equipment.

Monitoring

CloudWatch metrics are produced for each local network interface, just as they are for elastic network interfaces. For more information, see Monitor network performance for ENA settings on your EC2 instance in the *Amazon EC2 User Guide*.

MAC addresses

AWS provides MAC addresses for local network interfaces. Local network interfaces use locally administered addresses (LAA) for their MAC addresses. A local network interface uses the same MAC address until you delete the interface. After you delete a local network interface, remove the MAC address from your local configurations. AWS can reuse MAC addresses that are no longer in use.

Add a local network interface to an EC2 instance in an Outposts subnet

You can add a local network interface to an Amazon EC2 instance on an Outposts subnet during or after launch. You do so by adding a secondary network interface to the instance, using the device index that you specified when you enabled the Outpost subnet for local network interfaces.

Consideration

When you specify the secondary network interface using the console, the network interface is created using device index 1. If this is not the device index that you specified when you enabled the Outpost subnet for local network interfaces, you can specify the correct device index by using the AWS CLI or an AWS SDK instead. For example, use the following commands from the AWS CLI: create-network-interface and attach-network-interface.

Use the following procedure to add the local network interface after you launch the instance. For information about adding it during instance launch, see Launch an instance on the Outpost.

To add a local network interface to an EC2 instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network and Security**, **Network Interfaces**.
3. **Create the network interface**
    a. Choose **Create network interface**.
    b. Select the same Outpost subnet as the instance.
    c. Verify that **Private IPv4 address** is set to **Auto-assign**.
    d. Select any security group. Security groups do not apply to local network interface, so the security group that you select is not relevant.
    e. Choose **Create network interface**.
4. **Attach the network interface to the instance**
    a. Select the check box for the newly created network interface.
    b. Choose **Actions**, **Attach**.
    c. Choose the instance.

d. Choose **Attach**. The network interface is attached at device index 1. If you specified 1 as the device index for the local network interface for the Outpost subnet, this network interface is the local network interface for the instance.

View the local network interface

While the instance is in the running state, you can use the Amazon EC2 console to view both the elastic network interface and the local network interface for the instances in your Outpost subnet. Select the instance and choose the **Networking** tab.

The console displays a private IPv4 address for the local network interface from the subnet CIDR. This address is not the IP address of the local network interface, and it is not usable. However, this address is allocated from the subnet CIDR, so you must account for it in your subnet sizing. You must set the IP address for the local network interface within the guest operating system, either statically or through your DHCP server.

Configure the operating system

After you enable local network interfaces, Amazon EC2 instances will have two network interfaces, one of which is a local network interface. Ensure that you configure the operating system of the Amazon EC2 instances that you launch to support a multi-homed networking configuration.

Local network connectivity for Outposts servers

Use this topic to understand the network cabling and topology requirements for hosting an Outposts server. For more information, see Local network interfaces for your Outposts servers.

Contents

Server topology on your network

An Outposts server requires two distinct connections to your networking equipment. Each connection uses a different cable and carries a different type of traffic. The multiple cables are for traffic-class isolation only, and not for redundancy. The two cables do not need to connect to a common network.

The following table describes Outposts server traffic types and labels.

Traffic label	Description
2	**Service link traffic** – This traffic enables communication between the Outpost and the AWS Region for both management of the Outpost and intra-VPC traffic between the AWS Region and the Outpost. Service link traffic includes the service link connectio n from the Outpost to the Region. The service link is a custom VPN or VPNs from the Outpost to the Region. The Outpost connects to the Availability Zone in the Region that you chose at time of purchase.
1	**Local network interface link traffic** – This traffic enables communication from your VPC to your local LAN over the local network interface. Local link traffic includes instances running on the Outpost that communicate with your on-premises network. Local link traffic can also include instances communica ting with the internet through your on-premis es network.

Server physical connectivity

Each Outposts server includes non-redundant physical uplink ports. Ports have their own speed and connector requirements as follows:

- **10Gbe** – connector type QSFP+

QSFP+ cable

The QSFP+ cable has a connector that you attach to port 3 on the Outposts server. The other end of the QSFP+ cable has four SFP+ interfaces that you connect to your switch. Two of the switch-side interfaces are labeled 1 and 2. Both the interfaces are required for an Outposts server to function. Use the 2 interface for service link traffic and the 1 interface for local network interface link traffic. The remaining interfaces are not used.

Service link traffic for servers

Configure the service link port on your switch as an untagged access port to a VLAN with a gateway and a route to the following Region endpoints:

- Service link endpoints
- Outposts registration endpoint

The service link connection must have public DNS available for the Outpost to discover its registration endpoint in the AWS Region. The connection can have a NAT device between the Outposts server and the registration endpoint. For more information about the public address ranges for AWS, see AWS IP address ranges in the *Amazon VPC User Guide* and AWS Outposts endpoints and quotas in the *AWS General Reference*.

To register the server, open the following network ports:

- TCP 443
- UDP 443
- UDP 53

Local network interface link traffic

Configure the local network interface link port on your upstream network device as a standard access port to a VLAN on your local network. If you have more than one VLAN, configure all the ports on the upstream network device as trunk ports. Configure the port on your upstream network device to expect multiple MAC addresses. Each instance launched on the server will use

a MAC address. Some network devices offer port-security features that will shut down a port that reports multiple MAC addresses.

> **Note**
>
> AWS Outposts servers do not tag VLAN traffic. If you configure your local network interface as trunk, you must ensure that your OS tags VLAN traffic.

The following example shows how to configure VLAN tagging for your local network interface on Amazon Linux 2023. If you are using another Linux distribution, see the documentation for your Linux distribution about configuring VLAN tagging.

Example: To configure VLAN tagging for your local network interface on Amazon Linux 2023 and Amazon Linux 2

1. Ensure that the 8021q module is loaded into the kernel. If not, load it using the `modprobe` command.

```
modinfo 8021q
modprobe --first-time 8021q
```

2. Create the VLAN device. In this example:

   - The interface name of the local network interface is `ens6`
   - The VLAN id is 59
   - The name assigned for the VLAN device is `ens6.59`

```
ip link add link ens6 name ens6.59 type vlan id 59
```

3. Optional. Complete this step if you want to manually assign the IP. In this example we are assigning the IP 192.168.59.205, where the subnet CIDR is 192.168.59.0/24.

```
ip addr add 192.168.59.205/24 brd 192.168.59.255 dev ens6.59
```

4. Activate the link.

```
ip link set dev ens6.59 up
```

To configure your network interfaces at the OS level and make the VLAN tagging changes persistent, refer to the following resources:

- If you are using Amazon Linux 2, see Configure your network interface using ec2-net-utils for AL2 in the *Amazon Linux 2 User Guide*.
- If you are using Amazon Linux 2023, see Networking service in the *Amazon Linux 2023 User Guide*.

Server IP address assignment

You do not need public IP address assignments for the AWS Outposts server's service link and local network interfaces on instances. For the service link, you can assign IP addresses manually or use the Dynamic host control protocol (DHCP). To configure the service link connection, see Configure and test the connection in the *AWS Outposts server installation guide*.

To configure the local network interface link, see the section called "Configure the operating system".

> **Note**
>
> Ensure that you use a stable IP address for the Outposts server. IP address changes can cause temporary service disruptions on the Outpost subnet.

Server registration

When Outposts servers establish a connection on the local network, they use the service link connection to connect to Outpost registration endpoints and register themselves. Registration requires public DNS. When servers register, they create a secure tunnel to their service link endpoint in the Region. Outposts servers use TCP port 443 to facilitate communication with the Region over the public internet. Outposts servers do not support private connectivity through VPC.

Capacity management for AWS Outposts

An Outpost provides a pool of AWS compute and storage capacity at your site as a private extension of an Availability Zone in an AWS Region. Because the compute and storage capacity available in the Outpost is finite and determined by the size and number of assets that AWS installs at your site, you get to decide how much Amazon EC2, Amazon EBS, and Amazon S3 on AWS Outposts capacity you need to run your initial workloads, accommodate future growth, and to provide extra capacity to mitigate server failures and maintenance events.

Topics

View AWS Outposts capacity

You can view the capacity configuration at the instance or Outpost level.

To view capacity configuration for your Outpost using the console

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. From the left navigation pane, choose **Outposts**.
3. Choose the Outpost.
4. On the Outpost details page select either **Instance view** or **Rack view**.
   - **Instance view** - Provides information on the instances configured on the Outposts and the distribution of instances by size and family.
   - **Rack view** - Provides visualization of the instances on each asset within each Outpost and allows you to select **Modify instance capacity** to make changes to instance capacity.

Modify AWS Outposts instance capacity

The capacity of each new Outpost order is configured with a default capacity configuration. You can convert the default configuration to create various instances to meet your business needs. To

do so, you create a capacity task, choose an Outposts or a single asset, specify the instance sizes and quantity, and run the capacity task to implement the changes.

Considerations

Consider the following before modifying instance capacity:

- Capacity tasks can be run only by the AWS account that owns the Outpost resources (owner). Consumers cannot run capacity tasks. For more information about owners and consumers, see Share your AWS Outposts resources.
- Instances sizes and quantities can be defined at the Outpost level or at an individual asset level.
- Capacity is configured automatically across an asset or all the assets in an Outpost based on possible configurations and best practices.
- While a capacity task is running, the assets associated with the selected outpost may be isolated. For this reason we recommend creating a capacity task only when you don't expect to launch new instances on your Outposts.
- You can choose to run the capacity task instantly or to keep attempting periodically over the next 48 hours. Choosing to run instantly requires less asset isolation time, but the task might fail if instances need to be stopped to run the task. Choosing to run periodically allows more time to stop instances before the task would fail, but assets may be isolated for longer.
- It is possible for valid capacity configurations to not utilize all of the available vCPU on an asset. When this is the case, a message at the end of the **Instance type** section will inform you that you are under capacity, but will allow the configuration to be applied as requested.
- When you modify an Outpost in the console, not all supported instances are shown because mixing disk-backed instances with non-disk-backed instances is not fully supported in console. To access all possible instances, utilize the StartCapacityTask API.
- You can only modify your existing Outposts capacity configuration to use valid Amazon EC2 instance sizes from instance families supported on your respective asset model.
- If you have instances running on your Outpost that you do not want to stop to run a capacity task, select their respective Instance ID under the section **Instances to keep as-is – *optional*** and make sure to retain the necessary quantity of this instance size in your updated capacity configuration. This will retain instances being used to support production workloads while a capacity task runs.
- When configuring an asset with multiple instance sizes within an instance family, use **Auto-balance** to make sure you aren’t attempting to over or under-provision your droplet. Over-provisioning is not supported, and will cause a capacity task failure.

- Several capacity tasks can run in parallel as long as they apply to mutually exclusive sets of AssetIDs. For example, you can create several asset-level capacity tasks for different AssetIDs at the same time. However, if there is a running Outpost-level task, you cannot create another Outpost or asset-level task at the same time. Similarly, if there is a running asset-level task, you cannot create an Outpost-level task or an asset-level task on the same AssetID at the same time.

To modify capacity configuration for your Outpost using the console

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. From the left navigation pane, choose **Capacity tasks**.
3. On the **Capacity tasks** page, choose **Create capacity task**.
4. On the **Getting started** page, choose the order, Outpost, or asset to configure.
5. To modify capacity, specify an option for **Method of modification**: e steps in the console or upload a JSON file.

   - **Modify capacity configuration plan** to use the steps in the console
   - **Upload a capacity configuration plan** to upload a JSON file

> **Note**
>
> - To prevent capacity management from recommending specific instances to stop, specify the instances that should not be stopped. These instances will be excluded from the list of instances to stop.

Console steps

1. Choose **Instance view** or **Rack view**.
2. Choose **Modify an Outpost capacity configuration** or **Modify** on a single asset.
3. Choose an Outpost or asset if different than the current selection.
4. Choose to either run this capacity task immediately or periodically over 48 hours.
5. Choose **Next**.
6. On the **Configure instance capacity** page, each instance type shows one instance size with the maximum quantity preselected. To add more instance sizes, choose **Add instance size**.

7. Specify the instance quantity and note the capacity that is displayed for that instance size.
8. View the message at the end of each instance-type section that informs you if you are over or under capacity. Make adjustments at the instance size or quantity level to optimize your total available capacity.
9. You can also request AWS Outposts to optimize the instance quantity for a specific instance size. To do so:

   a. Choose the instance size.

   b. Choose **Auto-balance** at the end of the related instance-type section.
10. For each instance type, ensure that the instance quantity is specified for at least one instance size.
11. Optionally, choose instances to keep as-is.
12. Choose **Next**.
13. On the **Review and create** page, verify the updates that you are requesting.
14. Choose **Create**. AWS Outposts creates a capacity task.
15. On the capacity task page, monitor the status of the task.

Upload a JSON file

1. Choose **Upload a capacity configuration**.
2. Choose **Next**.
3. On the **Upload capacity configuration plan** page, upload the JSON file that specifies the instance type, size, and quantity. Optionally, you can specify the InstancesToExclude, and TaskActionOnBlockingInstances parameters in the JSON file.

   Example

   Example JSON file:

```
{
  "InstancePools": [
    {
      "InstanceType": "c5.24xlarge",
      "Count": 1
    },
    {
```

```
      "InstanceType": "m5.24xlarge",
      "Count": 2
    }
  ],
  "InstancesToExclude": {
    "AccountIds": [
      "111122223333"
    ],
    "Instances": [
      "i-1234567890abcdef0"
    ],
    "Services": [
      "ALB"
    ]
  },
  "TaskActionOnBlockingInstances": "WAIT_FOR_EVACUATION"
}
```

4. Review the contents of the JSON file in the **Capacity configuration plan** section.
5. Choose **Next**.
6. On the **Review and create** page, verify the updates that you are requesting.
7. Choose **Create**. AWS Outposts creates a capacity task.
8. On the capacity task page, monitor the status of the task.

Troubleshooting capacity task issues

Review the following known issues to resolve an issue related to capacity management in a new order. If you do not see your issue listed, contact Support.

Order *oo-xxxxxx* is not associated with Outpost ID *op-xxxxx*

This issue occurs when you use the AWS CLI or API to run the `StartCapacityTask` and the Outpost ID in the request does not match the Outpost ID in the order.

To resolve this issue:

1. Sign in to AWS.
2. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
3. From the navigation pane, choose **Orders**.

4. Select the order and verify that the order status is one of the following: `PREPARING`, `IN_PROGRESS`, or `ACTIVE`.
5. Note the Outpost ID in the order.
6. Enter the correct Outpost ID in the `StartCapacityTask` API request.

The capacity plan includes instance types that are not supported

This issue occurs when you use the AWS CLI or API to create or modify the capacity task and the request contains unsupported instances types.

To resolve this issue, use the console or CLI.

Use the console

1. Sign in to AWS.
2. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
3. From the navigation pane, choose **Capacity task**.
4. Use the **Upload a capacity configuration** option to upload a JSON with the same list of instance types.
5. The console displays an error message with the list of supported instance types.
6. Correct the request to remove the unsupported instance types.
7. Create or modify the capacity task on the console using the corrected JSON or use the CLI or API with this corrected list of instance types.

Use the CLI

1. Use the GetOutpostSupportedInstanceTypes command to see the list of supported instance types.
2. Create or modify the capacity task with the correct list of instance types.

No Outpost with Outpost ID *op-xxxxx*

This issue occurs when you use the AWS CLI or API to run the `StartCapacityTask` and the request contains an Outpost ID that is not valid for one of the following reasons:

- The Outpost is in a different AWS Region.

- You do not have permissions to this Outpost.
- The Outpost ID is incorrect.

To resolve this issue:

1. Note the AWS Region that you used in the `StartCapacityTask` API request.
2. Use the `ListOutposts` API action to get a list of Outposts that you own in the AWS Region.
3. Check if the Outpost ID is listed.
4. Enter the correct Outpost ID in the `StartCapacityTask` request.
5. If you do not find the Outpost ID, use the `ListOutposts` API action again to check if the Outpost exists in a different AWS Region.

Active CapacityTask cap-*XXXX* already found for Outpost op-*XXXX*

This issue occurs when you use the AWS Outposts console or API to run StartCapacityTask on an Outpost and there is already a running capacity task for the Outpost. A capacity task is considered running if it has any of the following statuses: `REQUESTED`, `IN_PROGRESS`, `WAITING_FOR_EVACUATION`, or `CANCELLATION_IN_PROGRESS`.

To resolve this issue, use the AWS Outposts console or CLI.

Use the console

1. Sign in to AWS.
2. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
3. From the navigation pane, choose **Capacity tasks**.
4. Ensure that there are no running capacity tasks for the OutpostId.
5. If there are running capacity tasks for the OutpostId, wait for them to terminate, or cancel them if desired.
6. When there no running capacity tasks for the requested OutpostId, retry your request to create the capacity task.

Use the CLI

1. Use the ListCapacityTasks command to find running capacity tasks for the Outpost.

2. Wait for all running capacity tasks to terminate, or cancel them if desired.
3. When there no running capacity tasks for the requested OutpostId, retry your request to create the capacity task.

Active CapacityTask cap-*XXXX* already found for Asset *XXXX* on Outpost op-XXXX

This issue occurs when you use the AWS Outposts console or API to run StartCapacityTask on an asset and there is already a running capacity task for the asset. A capacity task is considered running if it has any of the following statuses: `REQUESTED`, `IN_PROGRESS`, `WAITING_FOR_EVACUATION`, or `CANCELLATION_IN_PROGRESS`.

To resolve this issue, use the AWS Outposts console or CLI.

Use the console

1. Sign in to AWS.
2. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
3. From the navigation pane, choose **Capacity tasks**.
4. Ensure that there are no running capacity tasks for the OutpostId and no running asset-level capacity Tasks for the AssetId.
5. If there are running capacity tasks, wait for them to terminate, or cancel them if desired.
6. When there no running capacity tasks, retry your request to create the capacity task.

Use the CLI

1. Use the ListCapacityTasks command to find running capacity tasks for the OutpostID and AssetID.
2. Ensure that there are no running Outpost-level capacity tasks for the OutpostId, and no running asset-level capacity Tasks for the AssetId.
3. If there are running capacity tasks, wait for them to terminate, or cancel them if desired.
4. Retry your request to create the capacity task.

AssetId=*XXXX* is not valid for Outpost=op-*XXXX*

This issue occurs when you use the AWS Outposts console or API to run StartCapacityTask on an asset and the AssetID is not valid for one of the following reasons:

- The asset is not associated with the Outpost.
- The asset is isolated.

To resolve this issue, use the AWS Outposts console or CLI.

Use the console

1. Sign in to AWS.
2. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
3. Choose **Rack view** for the Outpost.
4. Verify that the requested AssetId is associated with the Outpost, and that it is not marked as an Isolated Host.
    a. If the Asset is isolated, this may be because a capacity task is running on it. You can navigate to the capacity tasks panel and check if there are any running Outpost or asset-level tasks for the OutpostId and AssetId. If there are, then wait for the task to terminate and for the asset to become available again.
    b. If there are no running capacity tasks for an isolated asset, then the asset may be degraded.
5. After you verify that the asset exists and is in a valid state, retry your request to create the capacity task.

Use the CLI

1. Use the ListAssets command to find the assets associated with the OutpostID.
2. Verify that the requested AssetId is associated with the Outpost, and that its State is `ACTIVE`.
    a. If the asset State is not ACTIVE, this may be because a capacity task is running on it. Use the ListCapacityTasks command to determine if there are running Outpost or asset-level tasks for the OutpostId and AssetId. If there are, then wait for the task to terminate and for the asset to become ACTIVE again.

b. If there are no running capacity tasks for an isolated asset, then the asset may be degraded.

3. After you verify that the asset exists and is in a valid state, retry your request to create the capacity task.

Share your AWS Outposts resources

With Outpost sharing, Outpost owners can share their Outposts and Outpost resources, including Outpost sites and subnets, with other AWS accounts under the same AWS organization. As an Outpost owner, you can create and manage Outpost resources centrally, and share the resources across multiple AWS accounts within your AWS organization. This allows other consumers to use Outpost sites, configure VPCs, and launch and run instances on the shared Outpost.

In this model, the AWS account that owns the Outpost resources (*owner*) shares the resources with other AWS accounts (*consumers*) in the same organization. Consumers can create resources on Outposts that are shared with them in the same way that they would create resources on Outposts that they create in their own account. The owner is responsible for managing the Outpost and resources that they create in it. Owners can change or revoke shared access at any time. With the exception of instances that consume Capacity Reservations, owners can also view, modify, and delete resources that consumers create on shared Outposts. Owners can't modify instances that consumers launch into Capacity Reservations that they have shared.

Consumers are responsible for managing the resources that they create on Outposts that are shared with them, including any resources that consume Capacity Reservations. Consumers can't view or modify resources owned by other consumers or by the Outpost owner. They also can't modify Outposts that are shared with them.

An Outpost owner can share Outpost resources with:

- Specific AWS accounts inside of its organization in AWS Organizations.
- An organizational unit inside of its organization in AWS Organizations.
- Its entire organization in AWS Organizations.

Contents

- Identifying a shared Outpost resource
- Shared Outpost resource permissions
- Billing and metering
- Limitations

Shareable Outpost resources

An Outpost owner can share the Outpost resources listed in this section with consumers.

For Outposts server resources, see Working with shared AWS Outposts resources.

These are the resources available for Outposts servers. For Outposts rack resources, see Working with shared AWS Outposts resources in the AWS Outposts User Guide for Outposts racks.

- **Allocated Dedicated Hosts** – Consumers with access to this resource can:
  - Launch and run EC2 instances on a Dedicated Host.
- **Outposts** – Consumers with access to this resource can:
  - Create and manage subnets on the Outpost.
  - Use the AWS Outposts API to view information about the Outpost.
- **Sites** – Consumers with access to this resource can:
  - Create, manage, and control an Outpost at the site.
- **Subnets** – Consumers with access to this resource can:
  - View information about subnets.
  - Launch and run EC2 instances in subnets.

  Use the Amazon VPC console to share an Outpost subnet. For more information, see Sharing a subnet in the *Amazon VPC User Guide*.

Prerequisites for sharing Outposts resources

- To share an Outpost resource with your organization or an organizational unit in AWS Organizations, you must enable sharing with AWS Organizations. For more information, see Enable Sharing with AWS Organizations in the *AWS RAM User Guide*.
- To share an Outpost resource, you must own it in your AWS account. You can't share an Outpost resource that has been shared with you.

- To share an Outpost resource, you must share it with an account that is within your organization.

Related services

Outpost resource sharing integrates with AWS Resource Access Manager (AWS RAM). AWS RAM is a service that enables you to share your AWS resources with any AWS account or through AWS Organizations. With AWS RAM, you share resources that you own by creating a *resource share*. A resource share specifies the resources to share, and the consumers with whom to share them. Consumers can be individual AWS accounts, organizational units, or an entire organization in AWS Organizations.

For more information about AWS RAM, see the *AWS RAM User Guide*.

Sharing across Availability Zones

To ensure that resources are distributed across the Availability Zones for a Region, we independently map Availability Zones to names for each account. This could lead to Availability Zone naming differences across accounts. For example, the Availability Zone `us-east-1a` for your AWS account might not have the same location as `us-east-1a` for another AWS account.

To identify the location of your Outpost resource relative to your accounts, you must use the *Availability Zone ID* (AZ ID). The AZ ID is a unique and consistent identifier for an Availability Zone across all AWS accounts. For example, `use1-az1` is an AZ ID for the `us-east-1` Region and it is the same location in every AWS account.

To view the IDs for the Availability Zones in your account

1. Navigate to the AWS RAM console in the AWS RAM console.
2. The AZ IDs for the current Region are displayed in the **Your AZ ID** panel on the right-hand side of the screen.

> **Note**
>
> Local gateway route tables are in the same AZ as their Outpost, so you do not need to specify an AZ ID for route tables.

Sharing an Outpost resource

When an owner shares an Outpost with a consumer, the consumer can create resources on the Outpost in the same way that they would create resources on Outposts that they create in their own account. Consumers with access to shared local gateway route tables can create and manage VPC associations. For more information, see Shareable Outpost resources.

To share an Outpost resource, you must add it to a resource share. A resource share is an AWS RAM resource that lets you share your resources across AWS accounts. A resource share specifies the resources to share, and the consumers with whom they are shared. When you share an Outpost resource using the AWS Outposts console, you add it to an existing resource share. To add the Outpost resource to a new resource share, you must first create the resource share using the AWS RAM console.

If you are part of an organization in AWS Organizations and sharing within your organization is enabled, you can grant consumers in your organization access from the AWS RAM console to the shared Outpost resource. Otherwise, consumers receive an invitation to join the resource share and are granted access to the shared Outpost resource after accepting the invitation.

You can share an Outpost resource that you own using the AWS Outposts console, AWS RAM console, or the AWS CLI.

To share an Outpost that you own using the AWS Outposts console

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Outposts**.
3. Select the Outpost, and then choose **Actions**, **View details**.
4. On the **Outpost summary** page, choose **Resource shares**.
5. Choose **Create resource share**.

You are redirected to the AWS RAM console to finish sharing the Outpost using the following procedure. To share a local gateway route table that you own, use the following procedure as well.

To share an Outpost or local gateway route table that you own using the AWS RAM console

See Creating a Resource Share in the *AWS RAM User Guide*.

To share an Outpost or local gateway route table that you own using the AWS CLI

Use the create-resource-share command.

Unsharing a shared Outpost resource

When you unshare your Outpost with a consumer, the consumer can no longer do the following:

- View the Outpost in the AWS Outposts console.
- Create new subnets on the Outpost.
- Create new Amazon EBS volumes on the Outpost.
- View the Outpost details and instance types using the AWS Outposts console or the AWS CLI.

Subnets, volumes, or instances that the consumer created during the shared period are not deleted and the consumer can continue to do the following:

- Access and modify these resources.
- Launch new instances on an existing subnet that the consumer created.

To prevent the consumer from accessing their resources and launching new instances on your Outpost, request that the consumer delete their resources.

When a shared local gateway route table is unshared, the consumer can no longer create new VPC associations to it. Any existing VPC associations that the consumer created remain associated with the route table. Resources in these VPCs can continue to route traffic to the local gateway. To prevent this, request that the consumer delete the VPC associations.

To unshare a shared Outpost resource that you own, you must remove it from the resource share. You can do this using the AWS RAM console or the AWS CLI.

To unshare a shared Outpost resource that you own using the AWS RAM console

See Updating a Resource Share in the *AWS RAM User Guide*.

To unshare a shared Outpost resource that you own using the AWS CLI

Use the disassociate-resource-share command.

Identifying a shared Outpost resource

Owners and consumers can identify shared Outposts using the AWS Outposts console and AWS CLI. They can identify shared local gateway route tables using the AWS CLI.

To identify a shared Outpost using the AWS Outposts console

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Outposts**.
3. Select the Outpost, and then choose **Actions**, **View details**.
4. On the **Outpost summary** page, view the **Owner ID** to identify the AWS account ID of the Outpost owner.

To identify a shared Outpost resource using the AWS CLI

Use the list-outposts and describe-local-gateway-route-tables commands. These commands return the Outpost resources that you own and Outpost resources that are shared with you. `OwnerId` shows the AWS account ID of the Outpost resource owner.

Shared Outpost resource permissions

Permissions for owners

Owners are responsible for managing the Outpost and resources that they create in it. Owners can change or revoke shared access at any time. They can use AWS Organizations to view, modify, and delete resources that consumers create on shared Outposts.

Permissions for consumers

Consumers can create resources on Outposts that are shared with them in the same way that they would create resources on Outposts that they create in their own account. Consumers are responsible for managing the resources that they launch onto Outposts that are shared with them. Consumers can't view or modify resources owned by other consumers or by the Outpost owner, and they can't modify Outposts that are shared with them.

Billing and metering

Owners are billed for Outposts and Outpost resources that they share. They are also billed for any data transfer charges associated with their Outpost's service link VPN traffic from the AWS Region.

There are no additional charges for sharing local gateway route tables. For shared subnets, the VPC owner is billed for VPC-level resources such as AWS Direct Connect and VPN connections, NAT gateways, and Private Link connections.

Consumers are billed for application resources that they create on shared Outposts, such as load balancers and Amazon RDS databases. Consumers are also billed for chargeable data transfers from the AWS Region.

Limitations

The following limitations apply to working with AWS Outposts sharing:

- Limitations for shared subnets apply to working with AWS Outposts sharing. For more information about VPC sharing limits, see Limitations in the *Amazon Virtual Private Cloud User Guide*.
- Service quotas apply per individual account.

Third-party block storage on Outposts servers

With Outposts servers, you can leverage existing data you're stored on third-party storage arrays. You can specify external block data volumes and external block boot volumes for your EC2 instances on Outposts. Using this integration, you can use external block data and boot volumes backed by third-party vendors, such as NetApp ONTAP and Pure FlashArray storage systems.

Considerations

- Available on Outposts racks and Outposts 2U servers. Not available on Outposts 1U servers.
- Available in all AWS Regions where AWS Outposts is available, except the AWS GovCloud (US) Regions.
- Available at no extra charge.
- You are responsible for the configuration and day-to-day management of the storage array. You also create and manage the external block volumes on the storage array. If you have issues with the hardware, software, or connectivity for the storage array, contact the third-party storage vendor.

External block data volumes

After you provision and configure block data volumes backed by a compatible third-party storage system, you can attach the volumes to your EC2 instances when you launch them. If you configure the volumes for multi-attach on the storage array, you can attach a volume to multiple EC2 instances.

Key steps

- You are responsible for establishing connectivity between the Outpost subnets and the local network through the local network interface.
- You use the management interface for the external storage array to create the volume. Then, you'll configure the initiator mapping by created a new Initiator Group and adding the iSCSI Qualified Name (IQN) of the target EC2 instance to this group. This associates the external block data volume with the EC2 instance.
- You add the external data volume when you launch the instance. You'll need the Initiator IQN, the target IP address, the port, and the IQN of the external storage array. For more information, see Launch an instance on the Outpost.

For more information, see Simplifying the use of third-party block storage with AWS Outposts.

External block boot volumes

Booting an EC2 instance on Outposts from external storage arrays provides a centralized, cost-effective, and efficient solution for on-premises workloads that depend on third-party storage. You can choose between the following options:

iSCSI SAN boot

Provides direct booting from the external storage array. Utilizes an AWS-provided iPXE helper AMI so that the instances can boot from a network location. When iPXE is combined with iSCSI, the EC2 instance treats the remote iSCSI target (the storage array) as a local disk. All read and write operations from the operating system are performed on the external storage array.

iSCSI or NVMe-over-TCP LocalBoot

Launches EC2 instances using a copy of the boot volume retrieved from the storage array, leaving the original source image unmodified. We launch a helper instance using a LocalBoot AMI. This helper instance copies the boot volume from the storage array to the instance store of the EC2 instance, and acts as an iSCSI initiator or NVMe-over-TCP host. Finally, the EC2 instance reboots using the local instance store volume.

Because instance store is temporary storage, the boot volume is deleted when the EC2 instance is terminated. Therefore, this option is suitable for read-only boot volumes, such as those used in virtual desktop infrastructure (VDI).

You can't boot EC2 Windows instances using NVMe-over-TCP LocalBoot. This is only supported using EC2 Linux instances.

For more information, see Deploying external boot volumes for use with AWS Outposts.

Security in AWS Outposts

Security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to AWS Outposts, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

For more information about security and compliance for AWS Outposts, see the AWS Outposts servers FAQ.

This documentation helps you understand how to apply the shared responsibility model when using AWS Outposts. It shows you how to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your resources.

Contents

Data protection in AWS Outposts

The AWS shared responsibility model applies to data protection in AWS Outposts. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the

AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties.

For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

Encryption at rest

With AWS Outposts, all data is encrypted at rest. The key material is wrapped to an external key stored in a removable device, the Nitro Security Key (NSK). The NSK is required to decrypt the data on your Outposts server.

Encryption in transit

AWS encrypts in-transit data between your Outpost and its AWS Region. For more information, see Connectivity through service link.

Data deletion

When you terminate an EC2 instance, the memory allocated to it is scrubbed (set to zero) by the hypervisor before it is allocated to a new instance, and every block of storage is reset.

Destroying the Nitro Security Key cryptographically shreds the data on your Outpost. For more information, see Cryptographically shred server data.

Identity and access management (IAM) for AWS Outposts

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be authenticated (signed in) and authorized (have permissions) to use AWS Outposts resources. You can use IAM for no additional charge.

Contents

- AWS Outposts policy examples
- Service-linked roles for AWS Outposts
- AWS managed policies for AWS Outposts

How AWS Outposts works with IAM

Before you use IAM to manage access to AWS Outposts, learn what IAM features are available to use with AWS Outposts.

IAM feature	AWS Outposts support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys (service-specific)	Yes
ACLs	No
ABAC (tags in policies)	Yes
Temporary credentials	Yes
Principal permissions	Yes
Service roles	No
Service-linked roles	Yes

Identity-based policies for AWS Outposts

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can

perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for AWS Outposts

To view examples of AWS Outposts identity-based policies, see AWS Outposts policy examples.

Policy actions for AWS Outposts

Supports policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of AWS Outposts actions, see Actions defined by AWS Outposts in the *Service Authorization Reference*.

Policy actions in AWS Outposts use the following prefix before the action:

```
outposts
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
    "outposts:action1",
    "outposts:action2"
]
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `List`, include the following action:

```
"Action": "outposts:List*"
```

Policy resources for AWS Outposts

Supports policy resources: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

Some AWS Outposts API actions support multiple resources. To specify multiple resources in a single statement, separate the ARNs with commas.

```
"Resource": [
    "resource1",
    "resource2"
]
```

To see a list of AWS Outposts resource types and their ARNs, see Resource types defined by AWS Outposts in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by AWS Outposts.

Policy condition keys for AWS Outposts

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of AWS Outposts condition keys, see Condition keys for AWS Outposts in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by AWS Outposts.

To view examples of AWS Outposts identity-based policies, see AWS Outposts policy examples.

ABAC with AWS Outposts

Supports ABAC (tags in policies): Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see Define permissions with ABAC authorization in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with AWS Outposts

Supports temporary credentials: Yes

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switch from a user to an IAM role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

Cross-service principal permissions for AWS Outposts

Supports forward access sessions (FAS): Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

Service-linked roles for AWS Outposts

Supports service-linked roles: Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing AWS Outposts service-linked roles, see Service-linked roles for AWS Outposts.

AWS Outposts policy examples

By default, users and roles don't have permission to create or modify AWS Outposts resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Create IAM policies (console) in the *IAM User Guide*.

For details about actions and resource types defined by AWS Outposts, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for AWS Outposts in the *Service Authorization Reference*.

Contents

Policy best practices

Identity-based policies determine whether someone can create, access, or delete AWS Outposts resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies*

that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.

- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Example: Using resource-level permissions

The following example uses resource-level permissions to grant permission to get information about the specified Outpost.

JSON

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "outposts:GetOutpost",
            "Resource": "arn:aws:outposts:us-east-1:111122223333:outpost/
op-1234567890abcdef0"
        }
    ]
}
```

The following example uses resource-level permissions to grant permission to get information about the specified site.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "outposts:GetSite",
            "Resource": "arn:aws:outposts:us-east-1:111122223333:site/
os-0abcdef1234567890"
        }
    ]
}
```

Service-linked roles for AWS Outposts

AWS Outposts uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a type of service role that is linked directly to AWS Outposts. AWS Outposts defines service-linked roles and includes all the permissions that it requires to call other AWS services on your behalf.

A service-linked role makes setting up your AWS Outposts more efficient because you don't have to manually add the necessary permissions. AWS Outposts defines the permissions of its service-linked roles, and unless defined otherwise, only AWS Outposts can assume its roles. The defined

permissions include the trust policy and the permissions policy, and that permissions policy can't be attached to any other IAM entity.

You can delete a service-linked role only after first deleting the related resources. This protects your AWS Outposts resources because you can't inadvertently remove permission to access the resources.

Service-linked role permissions for AWS Outposts

AWS Outposts uses the service-linked role named **AWSServiceRoleForOutposts_*OutpostID***. This role grants Outposts permissions to manage networking resources to enable private connectivity on your behalf. This role also allows Outposts to create and configure network interfaces, manage security groups, and attach interfaces to service link endpoint instances. These permissions are necessary for establishing and maintaining the secure, private connection between your on-premises Outpost and AWS services, ensuring reliable operation of your Outpost deployment.

The AWSServiceRoleForOutposts_*OutpostID* service-linked role trusts the following services to assume the role:

- `outposts.amazonaws.com`

Service-linked role policies

The AWSServiceRoleForOutposts_*OutpostID* service-linked role includes the following policies:

- AWSOutpostsServiceRolePolicy
- AWSOutpostsPrivateConnectivityPolicy_*OutpostID*

AWSOutpostsServiceRolePolicy

The `AWSOutpostsServiceRolePolicy` policy enables access to AWS resources managed by AWS Outposts.

This policy allows AWS Outposts to complete the following actions on the specified resources:

- Action: `ec2:DescribeNetworkInterfaces` on all AWS resources
- Action: `ec2:DescribeSecurityGroups` on all AWS resources
- Action: `ec2:CreateSecurityGroup` on all AWS resources

- Action: `ec2:CreateNetworkInterface` on all AWS resources

AWSOutpostsPrivateConnectivityPolicy_OutpostID

The `AWSOutpostsPrivateConnectivityPolicy_`*`OutpostID`* policy allows AWS Outposts to complete the following actions on the specified resources:

- Action: `ec2:AuthorizeSecurityGroupIngress` on all AWS resources that match the following condition:

```
{ "StringLike" : { "ec2:ResourceTag/outposts:private-connectivity-resourceId" :
 "OutpostID" }} and { "StringEquals" : { "ec2:Vpc" : "vpcArn" }}
```

- Action: `ec2:AuthorizeSecurityGroupEgress` on all AWS resources that match the following condition:

```
{ "StringLike" : { "ec2:ResourceTag/outposts:private-connectivity-resourceId" :
 "OutpostID" }} and { "StringEquals" : { "ec2:Vpc" : "vpcArn" }}
```

- Action: `ec2:CreateNetworkInterfacePermission` on all AWS resources that match the following condition:

```
{ "StringLike" : { "ec2:ResourceTag/outposts:private-connectivity-resourceId" :
 "OutpostID" }} and { "StringEquals" : { "ec2:Vpc" : "vpcArn" }}
```

- Action: `ec2:CreateTags` on all AWS resources that match the following condition:

```
{ "StringLike" : { "aws:RequestTag/outposts:private-connectivity-resourceId" :
 "{{OutpostId}}*"}}
```

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-linked role permissions in the *IAM User Guide*.

Create a service-linked role for AWS Outposts

You don't need to manually create a service-linked role. When you configure private connectivity for your Outpost in the AWS Management Console, AWS Outposts creates the service-linked role for you.

Edit a service-linked role for AWS Outposts

AWS Outposts does not allow you to edit the AWSServiceRoleForOutposts_*OutpostID* service-linked role. After you create a service-linked role, you can't change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Update a service-linked role in the *IAM User Guide*.

Delete a service-linked role for AWS Outposts

If you no longer require a feature or service that requires a service-linked role, we recommend that you delete that role. That way you avoid having an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

If the AWS Outposts service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

You must delete your Outpost before you can delete the AWSServiceRoleForOutposts_*OutpostID* service-linked role.

Before you begin, make sure that your Outpost is not being shared using AWS Resource Access Manager (AWS RAM). For more information, see Unsharing a shared Outpost resource.

To delete AWS Outposts resources used by the AWSServiceRoleForOutposts_*OutpostID*

Contact AWS Enterprise Support to delete your Outpost.

To manually delete the service-linked role using IAM

For more information, see Delete a service-linked role in the *IAM User Guide*.

Supported Regions for AWS Outposts service-linked roles

AWS Outposts supports using service-linked roles in all of the Regions where the service is available. For more information, see the FAQs for Outposts servers.

AWS managed policies for AWS Outposts

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

AWS managed policy: AWSOutpostsServiceRolePolicy

This policy is attached to a service-linked role that allows AWS Outposts to perform actions on your behalf. For more information, see Service-linked roles.

AWS managed policy: AWSOutpostsAuthorizeServerPolicy

Use this policy to grant the permissions required to authorize Outposts server hardware in your on-premises network.

This policy includes the following permissions.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "outposts:StartConnection",
        "outposts:GetConnection"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS Outposts updates to AWS managed policies

View details about updates to AWS managed policies for AWS Outposts since this service began tracking these changes.

Change	Description	Date
AWSOutpostsAuthorizeServerPolicy – New policy	AWS Outposts added a policy that grants permissions to authorize Outposts server hardware in your on-premises network.	January 4, 2023
AWS Outposts started tracking changes	AWS Outposts started tracking changes for its AWS managed policies.	December 03, 2019

Infrastructure security in AWS Outposts

As a managed service, AWS Outposts is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access AWS Outposts through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

For more information about the infrastructure security provided for the EC2 instances and EBS volumes running on your Outpost, see Infrastructure Security in Amazon EC2.

VPC Flow Logs function the same way as they do in an AWS Region. This means that they can be published to CloudWatch Logs, Amazon S3, or to Amazon GuardDuty for analysis. Data needs to be sent back to the Region for publication to these services, so it is not visible from CloudWatch or other services when the Outpost is in a disconnected state.

Resilience in AWS Outposts

For high availability, you can order additional Outposts servers. Outpost capacity configurations are designed to operate in production environments, and support N+1 instances for each instance family when you provision the capacity to do so. AWS recommends that you allocate sufficient additional capacity for your mission-critical applications to enable recovery and failover if there is an underlying host issue. You can use the Amazon CloudWatch capacity availability metrics and set alarms to monitor the health of your applications, create CloudWatch actions to configure automatic recovery options, and monitor the capacity utilization of your Outposts over time.

When you create an Outpost, you select an Availability Zone from an AWS Region. This Availability Zone supports control plane operations such as responding to API calls, monitoring the Outpost, and updating the Outpost. To benefit from the resiliency provided by Availability Zones, you can deploy applications on multiple Outposts, each attached to a different Availability Zone. This enables you to build additional application resilience and avoid a dependence on a single Availability Zone. For more information about Regions and Availability Zones, see AWS Global Infrastructure.

Outposts servers include instance store volumes but do not support Amazon EBS volumes. The data on instance store volumes persists after an instance reboot but does not persist after instance termination. To retain the long-term data on your instance store volumes beyond the lifetime of the instance, be sure to back up the data to persistent storage, such as an Amazon S3 bucket or a network storage device in your on-premises network.

Compliance validation for AWS Outposts

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security Compliance & Governance – These solution implementation guides discuss architectural considerations and provide steps for deploying security and compliance features.
- HIPAA Eligible Services Reference – Lists HIPAA eligible services. Not all AWS services are HIPAA eligible.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.
- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Monitor your Outposts server

AWS Outposts integrates with the following services that offer monitoring and logging capabilities:

CloudWatch metrics

Use Amazon CloudWatch to retrieve statistics about data points for your Outposts server as an ordered set of time series data, known as *metrics*. You can use these metrics to verify that your system is performing as expected. For more information, see CloudWatch metrics for Outposts servers.

CloudTrail logs

Use AWS CloudTrail to capture detailed information about the calls made to AWS APIs. You can store these calls as log files in Amazon S3. You can use these CloudTrail logs to determine such information as which call was made, the source IP address where the call came from, who made the call, and when the call was made.

The CloudTrail logs contain information about the calls to API actions for AWS Outposts. They also contain information for calls to API actions from services on an Outpost, such as Amazon EC2 and Amazon EBS. For more information, see Log API calls using CloudTrail.

VPC Flow Logs

Use VPC Flow Logs to capture detailed information about the traffic going to and from your Outpost and within your Outpost. For more information, see VPC Flow Logs in the *Amazon VPC User Guide*.

Traffic Mirroring

Use Traffic Mirroring to copy and forward network traffic from your Outposts server to out-of-band security and monitoring appliances. You can use the mirrored traffic for content inspection, threat monitoring, or troubleshooting. For more information, see the Amazon VPC Traffic Mirroring Guide.

AWS Health Dashboard

The AWS Health Dashboard displays information and notifications that are initiated by changes in the health of AWS resources. The information is presented in two ways: on a dashboard that shows recent and upcoming events organized by category, and in a full event log that shows all events from the past 90 days. For example, a connectivity issue on the service link would initiate

an event that would appear on the dashboard and event log, and remain in the event log for 90 days. A part of the AWS Health service, AWS Health Dashboard requires no setup and can be viewed by any user that is authenticated in your account. For more information, see Getting started with the AWS Health Dashboard.

CloudWatch metrics for Outposts servers

AWS Outposts publishes data points to Amazon CloudWatch for your Outposts. CloudWatch enables you to retrieve statistics about those data points as an ordered set of time series data, known as *metrics*. Think of a metric as a variable to monitor, and the data points as the values of that variable over time. For example, you can monitor the instance capacity available to your Outpost over a specified time period. Each data point has an associated timestamp and an optional unit of measurement.

You can use metrics to verify that your system is performing as expected. For example, you can create a CloudWatch alarm to monitor the `ConnectedStatus` metric. If the average metric is less than 1, CloudWatch can initiate an action, such as sending a notification to an email address. You can then investigate potential on-premises or uplink networking issues that might be impacting the operations of your Outpost. Common issues include recent on-premises network configuration changes to firewall and NAT rules, or internet connection issues. For `ConnectedStatus` issues, we recommend verifying connectivity to the AWS Region from within your on-premises network, and contacting AWS Support if the problem persists.

For more information about creating a CloudWatch alarm, see Using Amazon CloudWatch Alarms in the *Amazon CloudWatch User Guide*. For more information about CloudWatch, see the Amazon CloudWatch User Guide.

Contents

Metrics

The `AWS/Outposts` namespace includes the following categories of metrics.

Contents

Instance metrics

The following metrics are available for Amazon EC2 instances.

Metric	Dimension	Description
`InstanceFamilyCapacityAvailability`	`InstanceFamily` and `OutpostId`	The percentage of instance capacity available. This metric does not include capacity for any Dedicated Hosts configured on the Outpost. **Unit**: Percent **Maximum resolution**: 5 minutes **Statistics**: The most useful statistics are `Average` and pNN.NN (percentiles).
`InstanceFamilyCapacityUtilization`	`Account`, `InstanceFamily` , and `OutpostId`	The percentage of instance capacity in use. This metric does not include capacity for any Dedicated Hosts configured on the Outpost. **Unit**: Percent **Maximum resolution**: 5 minutes **Statistics**: The most useful statistics are `Average` and pNN.NN (percentiles).

Metric	Dimension	Description
InstanceTypeCapacityAvailability	InstanceType and OutpostId	The percentage of instance capacity available. This metric does not include capacity for any Dedicated Hosts configured on the Outpost. **Unit**: Percent **Maximum resolution**: 5 minutes **Statistics**: The most useful statistics are Average and pNN.NN (percentiles).
InstanceTypeCapacityUtilization	Account, InstanceType , and OutpostId	The percentage of instance capacity in use. This metric does not include capacity for any Dedicated Hosts configured on the Outpost. **Unit**: Percent **Maximum resolution**: 5 minutes **Statistics**: The most useful statistics are Average and pNN.NN (percentiles).

Metric	Dimension	Description
`UsedInstanceType_Count`	`Account`, `InstanceType` , and `OutpostId`	The number of instance types that are currently in use, including any instance types used by managed services such as Amazon Relational Database Service (Amazon RDS) or Application Load Balancer. This metric does not include capacity for any Dedicated Hosts configured on the Outpost. **Unit**: Count **Maximum resolution**: 5 minutes

Metric	Dimension	Description
`AvailableInstanceType_Count`	`InstanceType` and `OutpostId`	The number of available instance types. This metric includes the `AvailableReservedInstances` count. To determine the number of instances that you can reserve, subtract the `AvailableReservedInstances` count from the `AvailableInstanceType_Count` count. `Number of instances that you can reserve = AvailableInstanceType_Count - AvailableReservedInstances` This metric does not include capacity for any Dedicated Hosts configured on the Outpost. **Unit**: Count **Maximum resolution**: 5 minutes

Metric	Dimension	Description
`AvailableReservedInstances`	`InstanceType` and `OutpostId`	The number of instances that are available for launch into the compute capacity reserved using Capacity Reservations. This metric does not include Amazon EC2 Reserved Instances. This metric does not include the number of instances that you can reserve. To determine how many instances you can reserve, subtract the `AvailableReservedInstances` count from the `AvailableInstanceType_Count` count. `Number of instances that you can reserve = AvailableInstanceType_Count - AvailableReservedInstances` **Unit**: Count **Maximum resolution**: 5 minutes

Metric	Dimension	Description
UsedReservedInstances	InstanceType and OutpostId	The number of instances that are running in the compute capacity reserved using Capacity Reservations. This metric does not include Amazon EC2 Reserved Instances. **Unit**: Count **Maximum resolution**: 5 minutes
TotalReservedInstances	InstanceType and OutpostId	The total number of instances, running and available for launch, provided by the compute capacity reserved using Capacity Reservations. This metric does not include Amazon EC2 Reserved Instances. **Unit**: Count **Maximum resolution**: 5 minutes

Outposts metrics

The following metrics are available for your Outposts.

Metric	Dimension	Description
ConnectedStatus	OutpostId	The status of an Outpost's service link connection. If the

Metric	Dimension	Description
		average statistic is less than 1, the connection is impaired. **Unit**: Count **Maximum resolution**: 1 minute **Statistics**: The most useful statistic is `Average`.
`CapacityExceptions`	`InstanceType` and `OutpostId`	The number of insufficient capacity errors for instance launches. **Unit**: Count **Maximum resolution**: 5 minutes **Statistics**: The most useful statistics are `Maximum` and `Minimum`.

Metric dimensions

To filter the metrics for your Outpost, use the following dimensions.

Dimension	Description
`Account`	The account or service using the capacity.
`InstanceFamily`	The instance family.
`InstanceType`	The instance type.
`OutpostId`	The ID of the Outpost.

View CloudWatch metrics for your Outposts server

You can view the CloudWatch metrics for your Outposts server using the CloudWatch console.

To view metrics using the CloudWatch console

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Select the **Outposts** namespace.
4. (Optional) To view a metric across all dimensions, enter its name in the search field.

To view metrics using the AWS CLI

Use the following list-metrics command to list the available metrics.

```
aws cloudwatch list-metrics --namespace AWS/Outposts
```

To get the statistics for a metric using the AWS CLI

Use the following get-metric-statistics command to get statistics for the specified metric and dimension. CloudWatch treats each unique combination of dimensions as a separate metric. You can't retrieve statistics using combinations of dimensions that were not specially published. You must specify the same dimensions that were used when the metrics were created.

```
aws cloudwatch get-metric-statistics \
--namespace AWS/Outposts --metric-name InstanceTypeCapacityUtilization \
--statistics Average --period 3600 \
--dimensions Name=OutpostId,Value=op-01234567890abcdef
 Name=InstanceType,Value=c5.xlarge \
--start-time 2019-12-01T00:00:00Z --end-time 2019-12-08T00:00:00Z
```

Log AWS Outposts API calls using AWS CloudTrail

AWS Outposts is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures API calls for AWS Outposts as events. The calls captured include calls from the AWS Outposts console and code calls to the AWS Outposts API operations. Using the information collected by CloudTrail, you can determine the request that was

made to AWS Outposts, the IP address from which the request was made, when it was made, and additional details.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root user or user credentials.
- Whether the request was made on behalf of an IAM Identity Center user.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

CloudTrail is active in your AWS account when you create the account, and you automatically have access to the CloudTrail **Event history**. The CloudTrail **Event history** provides a viewable, searchable, downloadable, and immutable record of the past 90 days of recorded management events in an AWS Region. For more information, see Working with CloudTrail Event history in the *AWS CloudTrail User Guide*. There are no CloudTrail charges for viewing the **Event history**.

For an ongoing record of events in your AWS account past 90 days, create a trail or a CloudTrail Lake event data store.

CloudTrail trails

A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. All trails created using the AWS Management Console are multi-Region. You can create a single-Region or a multi-Region trail by using the AWS CLI. Creating a multi-Region trail is recommended because you capture activity in all AWS Regions in your account. If you create a single-Region trail, you can view only the events logged in the trail's AWS Region. For more information about trails, see Creating a trail for your AWS account and Creating a trail for an organization in the *AWS CloudTrail User Guide*.

You can deliver one copy of your ongoing management events to your Amazon S3 bucket at no charge from CloudTrail by creating a trail, however, there are Amazon S3 storage charges. For more information about CloudTrail pricing, see AWS CloudTrail Pricing. For information about Amazon S3 pricing, see Amazon S3 Pricing.

CloudTrail Lake event data stores

CloudTrail Lake lets you run SQL-based queries on your events. CloudTrail Lake converts existing events in row-based JSON format to Apache ORC format. ORC is a columnar storage format

that is optimized for fast retrieval of data. Events are aggregated into *event data stores*, which are immutable collections of events based on criteria that you select by applying advanced event selectors. The selectors that you apply to an event data store control which events persist and are available for you to query. For more information about CloudTrail Lake, see Working with AWS CloudTrail Lake in the *AWS CloudTrail User Guide*.

CloudTrail Lake event data stores and queries incur costs. When you create an event data store, you choose the pricing option you want to use for the event data store. The pricing option determines the cost for ingesting and storing events, and the default and maximum retention period for the event data store. For more information about CloudTrail pricing, see AWS CloudTrail Pricing.

AWS Outposts management events in CloudTrail

Management events provide information about management operations that are performed on resources in your AWS account. These are also known as control plane operations. By default, CloudTrail logs management events.

AWS Outposts logs all AWS Outposts control plane operations as management events. For a list of the AWS Outposts control plane operations that AWS Outposts logs to CloudTrail, see the AWS Outposts API Reference.

AWS Outposts event examples

The following example shows a CloudTrail event that demonstrates the `SetSiteAddress` operation.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAIOSFODNN7EXAMPLE:jdoe",
        "arn": "arn:aws:sts::111122223333:assumed-role/example/jdoe",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/example",
```

```
                "accountId": "111122223333",
                "userName": "example"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-08-14T16:28:16Z"
            }
        }
    },
    "eventTime": "2020-08-14T16:32:23Z",
    "eventSource": "outposts.amazonaws.com",
    "eventName": "SetSiteAddress",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "XXX.XXX.XXX.XXX",
    "userAgent": "userAgent",
    "requestParameters": {
        "SiteId": "os-123ab4c56789de01f",
        "Address": "***"
    },
    "responseElements": {
        "Address": "***",
        "SiteId": "os-123ab4c56789de01f"
    },
    "requestID": "1abcd23e-f4gh-567j-klm8-9np01q234r56",
    "eventID": "1234a56b-c78d-9e0f-g1h2-34jk56m7n890",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Outposts server maintenance

Under the shared responsibility model, AWS is responsible for the hardware and software that run AWS services. This applies to AWS Outposts, just as it does to an AWS Region. For example, AWS manages security patches, updates firmware, and maintains the Outpost equipment. AWS also monitors the performance, health, and metrics for your Outposts server and determines whether any maintenance is required.

> **Warning**
>
> Data on instance store volumes is lost if the underlying disk drive fails, or if the instance terminates. To prevent data loss, we recommend that you back up your long-term data on instance store volumes to persistent storage, such as an Amazon S3 bucket or a network storage device in your on-premises network.

Contents

Update contact details

If the Outpost owner changes, contact AWS Support Center with the new owner's name and contact information.

Hardware maintenance

If AWS detects an irreparable issue with hardware during the server provisioning process or while hosting Amazon EC2 instances running on your Outposts server, we will notify the Outpost owner and the owner of the instances that the affected instances are scheduled for retirement. For more information, see Instance retirement in the *Amazon EC2 User Guide*.

AWS terminates the affected instances on the instance retirement date. The data on instance store volumes does not persist after instance termination. Therefore, it is important that you take action before the instance retirement date. First, transfer your long-term data from the instance store volumes for each affected instance to persistent storage, such as an Amazon S3 bucket or a network storage device in your network.

A replacement server will be shipped to the Outpost site. Then, do the following:

- Remove the network and power cables from the irreparable server and if necessary remove it from your rack.
- Install the replacement server in the same location. Follow the installation instructions in Outposts server installation.
- Pack the irreparable server to AWS in the same packaging that the replacement server arrived in.
- Use the pre-paid return shipment label that is available in the console attached to the order configuration details or the replacement server order.
- Return the server to AWS. For more information, see Return an AWS Outposts server.

Firmware updates

Updating the Outpost firmware does not typically affect the instances on your Outpost. In the rare case that we need to reboot the Outpost equipment to install an update, you will receive an instance retirement notice for any instances running on that capacity.

Best practices for power and network events

As stated in the AWS Service Terms for AWS Outposts customers, the facility where the Outposts equipment is located must meet the minimum power and network requirements to support the installation, maintenance, and use of the Outposts equipment. An Outposts server can operate correctly only when power and network connectivity is uninterrupted.

Power events

With complete power outages, there is an inherent risk that an AWS Outposts resource may not return to service automatically. In addition to deploying redundant power and backup power solutions, we recommend that you do the following in advance to mitigate the impact of some of the worst-case scenarios:

- Move your services and applications off the Outposts equipment in a controlled fashion, using DNS-based or off-rack load-balancing changes.
- Stop containers, instances, databases in an ordered incremental fashion and use the reverse order when restoring them.
- Test plans for the controlled moving or stopping of services.
- Back-up critical data and configurations and store it outside the Outposts.
- Keep power downtimes to a minimum.
- Avoid repeated switching of the power feeds (off-on-off-on) during the maintenance.
- Allow for extra time within the maintenance window to deal with the unexpected.
- Manage the expectations of your users and customers by communicating a wider maintenance window time-frame than you would normally need.
- After power is restored, create a case at AWS Support Center to request verification that AWS Outposts and the related services are running.

Network connectivity events

The service link connection between your Outpost and the AWS Region or Outposts home Region will typically automatically recover from network interruptions or issues that may occur in your upstream corporate network devices or in the network of any third party connectivity provider once the network maintenance is completed. During the time the service link connection is down, your Outposts operations are limited to local network activities.

Amazon EC2 instances, LNI networking, and instance storage volumes on the Outposts server will continue to operate normally and can be accessed locally through the local network and LNI. Similarly, AWS service resources such as Amazon ECS worker nodes continue to run locally. However, API availability will be degraded. For example, the run, start, stop, and terminate APIs might not work. Instance metrics and logs will continue to be cached locally for up to 7 days, and will be pushed to the AWS Region when connectivity returns. Disconnection beyond 7 days might result in loss of metrics and logs.

If the service link is down because of an on-site power issue or the loss of network connectivity, the AWS Health Dashboard sends a notification to the account that owns the Outposts. Neither you nor AWS can suppress the notification of a service link interruption, even if the interruption is expected. For more information, see Getting started with your AWS Health Dashboard in the *AWS Health User Guide*.

In the case of a planned service maintenance that will affect network connectivity, take the following proactive steps to limit the impact of potential problematic scenarios:

- If you are in control of the network maintenance, limit the duration of downtime for the service link. Include a step in your maintenance process that verifies that the network has recovered.
- If you are not in control of the network maintenance, monitor the service link downtime with respect to the announced maintenance window and escalate early to the party in charge of the planned network maintenance if the service link is not back up at the end of the announced maintenance window.

Resources

Here are some monitoring related resources that can provide reassurance that the Outposts is operating normally after a planned or unplanned power or network event:

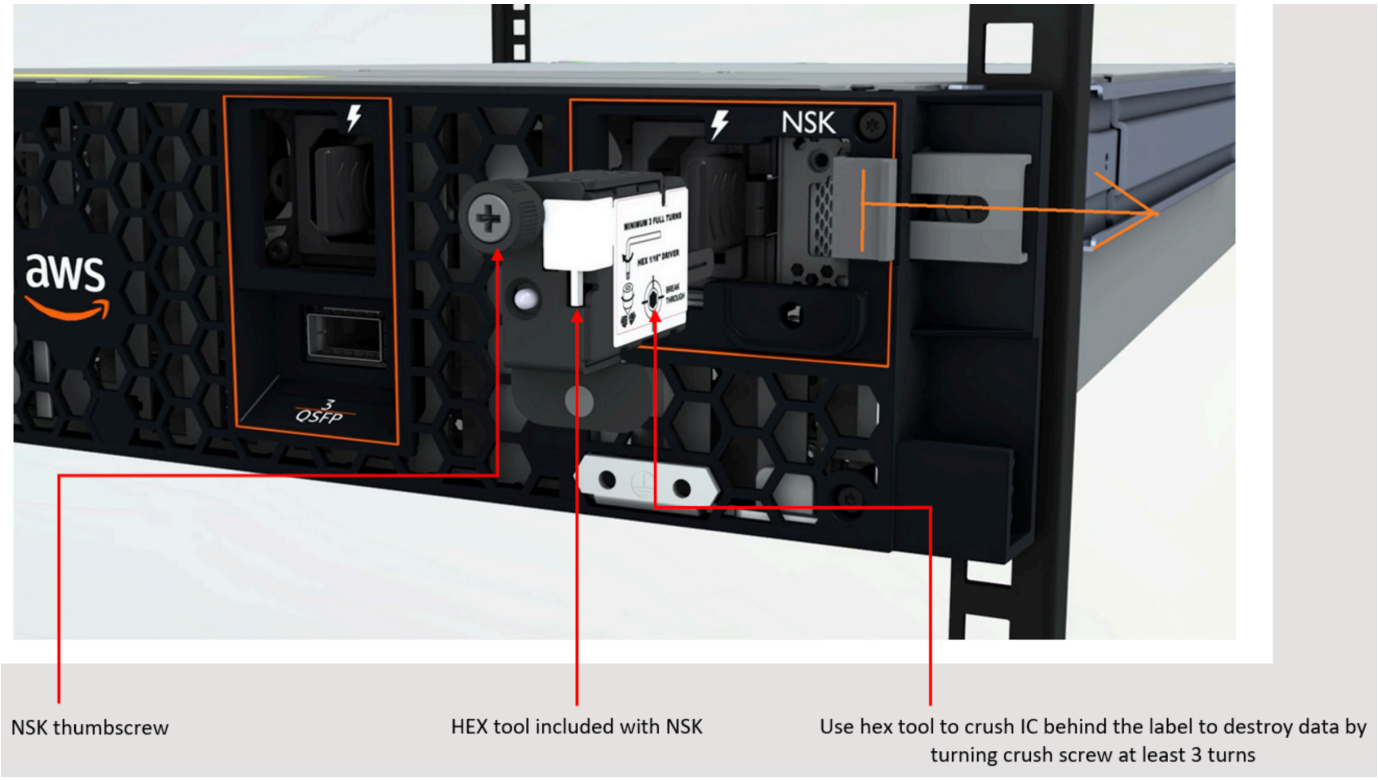
- The AWS blog Monitoring best practices for AWS Outposts covers observability and event management best practices specific to Outposts.
- The AWS blog Debugging tool for network connectivity from Amazon VPC explains the *AWSSupport-SetupIPMonitoringFromVPC* tool. This tool is an AWS Systems Manager document (SSM document) that creates an Amazon EC2 Monitor Instance in a subnet specified by you and monitors target IP addresses. The document runs ping, MTR, TCP trace-route and trace-path diagnostic tests and stores the results in Amazon CloudWatch Logs which can be visualized in a CloudWatch dashboard (e.g. latency, packet loss). For Outposts monitoring, the Monitor Instance should be in one subnet of the parent AWS Region and configured to monitor one or more of your Outpost instances using its private IP(s) - this will provide packet loss graphs and latency between AWS Outposts and the parent AWS Region.
- The AWS blog Deploying an automated Amazon CloudWatch dashboard for AWS Outposts using AWS CDK describes the steps involved in deploying an automated dashboard.
- If you have questions or need more information, see Creating a support case in the *AWS Support User Guide*.

Cryptographically shred server data

The Nitro Security Key (NSK) is required to decrypt data on the server. When you return the server to AWS, either because you are replacing the server or discontinuing the service, you can destroy the NSK to cryptographically shred the data on the server.

To cryptographically shred data on the server

1. Remove the NSK from the server before shipping the server back to AWS.
2. Ensure that you have the correct NSK that shipped with the server.
3. Remove the small hex tool / Allen wrench from under the sticker.
4. Use the hex tool to turn the small screw under the sticker three full turns. This action destroys the NSK and cryptographically shreds all data on the server.



Outposts server end-of-term options

At the end of your AWS Outposts term, you must choose between the following options:

- Renew your subscription and keep your existing Outposts servers.
- Return your Outposts servers.
- Convert to a month-to-month subscription and keep your existing Outposts servers.

Renew your subscription

You must complete the following steps at least **5 business days** before the current subscription for your Outposts servers ends. Failing to complete these steps at least 5 business days before the current subscription ends might result in unanticipated charges.

To renew your subscription and keep your existing Outposts servers

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. In the navigation pane, choose **Outposts**.
3. Choose **Actions**.
4. Choose **Renew Outpost**.
5. Choose the subscription term length and payment option.

   For pricing, see AWS Outposts servers pricing. You can also request a price quote.
6. Choose **Submit support ticket**.

> **Note**
>
> If renewing before the current subscription for your Outposts servers ends, you will be charged immediately for any upfront fees.

Your new subscription will start the day after your current subscription ends.

If you do not indicate that you want to renew your subscription or return your Outposts server, you will be converted to a month-to-month subscription automatically. Your Outpost will be

renewed on a monthly basis at the rate of the **No Upfront** payment option that corresponds to your AWS Outposts configuration. Your new monthly subscription will start the day after your current subscription ends.

Return Outposts servers

To return a server because the server reached the end of the contract term, you must first complete the decommission process at least **5 business days** before the current subscription for your Outposts servers ends. AWS can't start the return process until you do so. Failing to complete the decommission process at least 5 business days before the current subscription ends might result in delays in decommissioning and unanticipated charges.

After you complete the decommission process, you must prepare the server for return, obtain the shipping label, and pack and return the server to AWS.

You will not be charged a shipping fee when you return an Outposts server. However, if you return a server that is damaged, you might incur a cost.

Tasks

- Step 1: Prepare the server for return
- Step 2: Decommission the server
- Step 3: Obtain the return shipping label
- Step 4: Pack the server
- Step 5: Return the server through the courier

Step 1: Prepare the server for return

To prepare the server for return, unshare resources, backup data, delete local network interfaces and terminate active instances.

1. If the Outpost's resources are shared, you must unshare these resources.

   You can unshare a shared Outpost resource in one of the following ways:

   - Use the AWS RAM console. For more information, see Updating a resource share in the *AWS RAM User Guide*.
   - Use the AWS CLI to run the disassociate-resource-share command.

For the list of Outpost resources that can be shared, see Shareable Outpost resources.

2. Create backups of the data stored in the instance storage of the Amazon EC2 instances running on the AWS Outposts server.
3. Delete the local network interfaces associated with the instances that were running on the server.
4. Terminate the active instances associated with subnets on your Outpost. To terminate the instances, follow the instructions in Terminate your instance in the *Amazon EC2 User Guide*.
5. Destroy the Nitro Security Key (NSK) to cryptographically shred your data on the server. To destroy the NSK, follow the instructions in Cryptographically shred server data.

Step 2: Decommission the server

Complete the following steps at least **5 business days** before the current subscription for your Outposts servers ends.

> **Important**
>
> AWS can't stop the return process after you have submitted your decommission request.

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. In the navigation pane, choose **Outposts**.
3. Choose **Actions**.
4. Choose **Decommission Outpost**.
5. Choose a reason for the decommission.
6. Choose **Submit support ticket**.

> **Note**
>
> Returning your Outposts servers before the current subscription ends will not terminate any outstanding charges associated with this Outpost.

Step 3: Obtain the return shipping label

> **Important**
>
> You must only use the shipping label that AWS provides because it contains specific information, such as the Asset ID, about the server that you are returning. Do not create your own shipping label.

To obtain your shipping label:

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Orders**.
3. Under **Replacement order summary**, choose **Print return label** and choose the configuration ID of the server that you plan to return.

> **Note**
>
> Returning your Outposts servers before the current subscription ends will not terminate any outstanding charges associated with this Outpost.

Step 4: Pack the server

To pack your server, use the box and packaging material provided by AWS.

1. Pack the server in one of the following boxes:

   - The box and packaging material that the server originally came in.
   - The box and packaging material that the replacement server came in.

   Alternatively, contact AWS Support Center to request a box.
2. Affix the shipping label that AWS provided, to the outside of the box.

> **Important**
>
> Verify that the Asset ID on the shipping label matches the Asset ID on the server that you are returning.
> The Asset ID is located on the pull-out tab on the front of the server. Example: `1203779889` or `9305589922`

3. Seal the box securely.

Step 5: Return the server through the courier

You must return the server through the designated courier for your country. You can deliver the server to the courier or schedule the day and time that you prefer for the courier to pick up the server. The shipping label that AWS provides contains the correct address to return the server.

The following table shows who to contact for the country you are shipping from:

Country	Contact
Argentina	Contact AWS Support Center. In your request, include the following information: • The tracking number that is on the AWS-provided shipping label • The date and time that you prefer the courier to pick up the server • A contact name • A phone number • An email address
Bahrain	
Brazil	
Brunei	
Canada	
Chile	
Colombia	
Hong Kong	
India	
Indonesia	

Country	Contact
Japan	
Malaysia	
Nigeria	
Oman	
Panama	
Peru	
Philippines	
Serbia	
Singapore	
South Africa	
South Korea	
Taiwan	
Thailand	
United Arab Emirates	
Vietnam	

Country	Contact
United States of America	Contact UPS. You can return the server in the following ways: • Return the server during a routine UPS pickup at your site. • Drop-off the server at a UPS location. • Schedule a pickup for a date and time you prefer. Enter the tracking number from the AWS-provided shipping label for free shipping.
All other countries	Contact DHL. You can return the server in the following ways: • Drop-off the server at a DHL location. • Schedule a pickup for a date and time you prefer. Enter the DHL Waybill number from the AWS-provided shipping label for free shipping. If you get the following error `Courier pickup can't be scheduled for an import shipment`, it usually means that the pickup country that you selected does not match the pickup country on the return shipment label. Select the country where the shipment originates from and try again.

Convert to a month-to-month subscription

To convert to a month-to-month subscription and keep your existing Outposts servers, no action is needed. If you have questions, open a billing support case.

Your Outpost will be renewed on a monthly basis at the rate of the **No Upfront** payment option that corresponds to your AWS Outposts configuration. Your new monthly subscription starts the day after your current subscription ends.

Quotas for AWS Outposts

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is Region-specific. You can request increases for some quotas, but not for all quotas.

To view the quotas for AWS Outposts, open the Service Quotas console. In the navigation pane, choose **AWS services**, and select **AWS Outposts**.

To request a quota increase, see Requesting a quota increase in the *Service Quotas User Guide*.

Your AWS account has the following quotas related to AWS Outposts.

Resource	Default	Adjustable	Comments
Outpost sites	100	Yes	An Outpost site is the customer managed physical building where you power and attach your Outpost equipment to the network. You can have 100 Outposts sites in each Region of your AWS account.
Outposts per site	10	Yes	AWS Outposts includes hardware and virtual resources, known as Outposts. This quota limits your Outpost virtual resources. You can have 10 Outposts in each Outpost site.

AWS Outposts and the quotas for other services

AWS Outposts relies on the resources of other services and those services may have their own default quotas. For example, your quota for local network interfaces comes from the Amazon VPC quota for network interfaces.

Document history for Outposts servers

The following table describes the documentation updates for Outposts servers.

Change	Description	Date
Renewing your subscription and preparing servers for return	To renew a subscription or return a server, you must complete the process at least 10 business days before the current subscription ends.	July 16, 2025
Troubleshooting the service link connection	If the connection between your Outposts server and AWS Region is down, follow these steps to troubleshoot and resolve.	May 5, 2025
Updates to static stability	In the event that your network is interrupted, instance metrics and logs will be cached locally for up to 7 days. Previously, Outposts could cache logs for just a few hours.	May 1, 2025
Capacity management at the asset level	You can modify capacity configuration at the asset level.	March 31, 2025
External block volumes backed by third-party storage	You can now attach block data volumes backed by compatible third-party block storage systems during the instance-launch process on Outpost.	December 1, 2024

Capacity management	You can modify the default capacity configuration for your new Outposts order.	April 16, 2024
End-of-term options for AWS Outposts servers	At the end of your AWS Outposts term, you can renew, end, or convert your subscription.	August 1, 2023
Created AWS Outposts User Guide for Outposts servers	AWS Outposts User Guide broke into separate guides for rack and servers.	September 14, 2022
Placement groups on AWS Outposts	Placement groups that use a spread strategy can distribute instances across hosts.	June 30, 2022
Introducing Outposts servers	Added Outposts servers, a new AWS Outposts form factor.	November 30, 2021